P. 1877/93

2 1993

informatyka

WYDAWNICTWO SIGMA NOT

Cena 25 000 zł

informatyka

nr 2 1993
Luty
Rok wydania XXVIII

Miesięcznik
ISSN 0542-9951
INDEKS 36124

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
mgr inż. Piotr FUGLEWICZ
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
mgr inż. Jan RYZKO
dr Zdzisław SZYJEWSKI

PRZEWODNICZĄCY
RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:
01-552 Warszawa,
Pl. Inwalidów 10, p. 104, 105
tel. 39-14-34

Materiałów nie zamówionych
redakcja nie zwraca

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji lub Działu Reklamy i Marketingu 00-950 Warszawa ul. Mazowiecka 12 telefon: 27-43-66 telefaks: 19-21-87 teleks: 814877

W numerze:

Strona

Systemy ekspertowe w zarządzaniu – *Piotr Sienkiewicz* 1
Informacyjna infrastruktura zarządzania – potrzeby i tendencje zmian – *Jerzy Kisielnicki* 8
Języki czwartej generacji a systemy CASE – *Piotr Fuglewicz* 11
Zastosowania wielofunkcyjnych elektronicznych kart bezstykowych – *Andrzej Bendig-Wielowiejski, Andrzej Ruciński, Jerzy Sarad* 13
TIFF – standard grafiki pikselowej – *Jarosław A. Berent* 16
Matematyka w tworzeniu oprogramowania? – *Jarosław Bochiński* 18
Podstawowe problemy rozpoznawania pisma drukowanego w systemach off-line – *Iwona Dąbrowska, Dorota Depowska* 22

RECENZJE
Algorytmika – sedno informatyki – *Zbigniew Weiss* 26

W najbliższych numerach:

- **Anna Ostaszewska i Piotr Fuglewicz podejmują próbę scharakteryzowania systemu zarządzania bazą danych PROGRESS oraz porównują ją z innymi bazami danych.**
- **Zygmunt Drążek omawia środowisko programowe do budowy systemów wspomagania decyzji na podstawie systemu FCS.**
- **Marek Miłosz zajmuje się mikrokomputerową symulacją systemów z wykorzystaniem języka GPSS.**
- **Magdalena Bijałd, Marcin Gwóźdź i Krzysztof Zieliński charakteryzują przetwarzanie rozproszone w systemach otwartych, na podstawie systemu ANSA, oraz omawiają jego podstawowe założenia.**
- **Waldemar Wieczerzycki omawia problem zarządzania współbieżnym wykonywaniem transakcji w celu zagwarantowania spójności bazy danych.**
- **Andrzej Matlak naświetla sytuację w USA w zakresie prawnych środków ochrony programów komputerowych oraz charakteryzuje możliwe kierunki zmian w tym zakresie.**
- **Zygmunt Topolewski zajmuje się szyfrowaniem informacji, a ściślej – kluczami szyfrującymi w metodzie RSA.**

Warunki prenumeraty na 1993 r.

Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

Zamawiający może otrzymywać zaprenumerowany przez siebie tytuł począwszy od następnego miesiąca po dokonaniu wpłaty. Zamówienia na zeszyty sprzed daty otrzymania wpłaty będą realizowane w miarę możliwości – z posiadanych zapasów magazynowych.

Warunkiem przyjęcia i realizacji zamówienia jest otrzymanie z banku potwierdzenia dokonania wpłaty przez prenumeratora. Dokument wpłaty jest równoznaczny ze złożeniem zamówienia.

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując środki pod adres:
Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-950 Warszawa, skr. poczt. 1004
konto:
PBK III 0/Warszawa nr 370015-1573-139-11

Na blankiecie wpłaty należy czytelnie podać nazwę zamawianego czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty oraz własny adres.
Na życzenie prenumeratora, zgłoszone np. telefonicznie, Zakład Kolportażu ul. Bartycka 20, 00-950 Warszawa, (telefony: 40-30-86, 40-35-89 oraz 40-00-21 wew. 249, 293, 299) wysyła specjalne blankiety zamówień wraz z aktualną listą tytułów i cennikiem czasopism.

Odbiorcy zagraniczni mogą otrzymywać czasopisma poprzez prenumeratę dewizową (wpłata dokonywana poza granicami Polski w dewizach, wg cennika dewizowego z cenami podanymi w dolarach amerykańskich) lub poprzez zamówioną w kraju prenumeratę ze zleceniem wysyłki za granicę (zamawiający podaje dokładny adres odbiorcy za granicą, dokonując równocześnie wpłaty w wysokości dwukrotnie wyższej niż cena normalnej prenumeraty krajowej).

Egzemplarze archiwalne (sprzedaż przelewowa lub za zaliczeniem pocztowym) można zamawiać pisemnie, kierując zamówienia pod adresem: Wydawnictwo SIGMA NOT, Spółka z o.o. Zakład Kolportażu, 00-716 Warszawa, ul. Bartycka 20, paw. B, tel. 40-37-31, natomiast za gotówkę można je nabyć w Klubie Prasy Technicznej w Warszawie ul. Mazowieckiej 12, tel. 26-80-17.

Istnieje możliwość zaprenumerowania 1 egz. czasopisma po cenie ulgowej przez indywidualnych członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT oraz przez uczniów zawodowych i studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT lub szkoły.

W przypadku zmiany cen w okresie objętym prenumeratą Wydawnictwo zastrzega sobie prawo do wystąpienia o dopłatę różnicy cen oraz prawo do realizowania prenumeraty tylko w pełni opłaconej.
Cena jednego egzemplarza: normalna 25 000 zł, ulgowa 18 750 zł
Wartość prenumeraty w zł:
Normalna: kwartalna 75 000, półroczna 150 000, roczna 300 000
Ulgowa: kwartalna 56 250, półroczna 112 500, roczna 225 000.

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 27/93

Artykuł ten, a także trzy następne, pochodzą z materiałów Konferencji INFOGRYF'92, jaka odbyła się w październiku ub.r. w Kołobrzegu.

PIOTR SIENKIEWICZ
Akademia Obrony Narodowej
Warszawa

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P. 1877/93

Systemy ekspertowe w zarządzaniu

Najczęstszy ludzki błąd
– nie przewidzieć burzy w piękny czas
Niccolo Machiavelli

Spośród wielu określeń istoty zarządzania na szczególną uwagę zasługują te, gdzie szczególną wagę przywiązuje się do procesu decyzyjnego, który ma zapewnić eliminację wykrytych zagrożeń, wykorzystanie zarysowujących się szans oraz efektywne spełnianie przez organizację wszystkich funkcji niezbędnych do osiągnięcia zamierzonych celów. Podejmowanie decyzji w zarządzaniu wiąże się z analizą sytuacji systemowych w rezultacie której dokonuje się oceny szans i zagrożeń, niezbędnej do określenia dopuszczalnych wariantów działania. Faza stricte decyzyjna wiąże się z koniecznością dokonania wyboru wariantu optymalnego (lub zadowalającego) w sensie przyjętych kryteriów efektywności organizacji oraz preferencji decydenta (decydentów).

Dziedzinami zarządzania są zasoby ludzkie, zasoby fizyczne, pieniądze i czas. Uważa się, że efektywnie działający szefowie przedsiębiorstw zawdzięczali swój sukces innowacyjności i zdolności do improwizacji zmierzających do zaspokojenia zmieniających się upodobań i wymagań mody oraz z uwzględnieniem posiadanych przez klientów środków.

Lee Laccoca wspomina:
„Gdybym miał zrekapitulować w dwu słowach cechy dobrego menadżera, powiedziałbym, że wszystko sprowadza się do zdolności decyzyjnych. Można wykorzystywać liczby, ale w końcu trzeba jednak uzyskane informacje poddać syntezie i ustalić harmonogram – a potem działać. Nie radzę jednak działać pochopnie. (...) Ilekroć podejmuję ryzyko, muszę przedtem upewnić się, że wyniki badań naukowych i analizy rynku potwierdzają trafność mojego wyczucia. Jeśli działam na podstawie intuicji, to tylko wtedy, gdy przeczucia są poparte faktami" [5].

Sprowadzanie więc zarządzania do rozwiązywania „ciągu" problemów (zadań) stricte decyzyjnych jest pewną idealizacją. Zapewne, obok programowanych i nieprogramowanych zadań decyzyjnych – o różnym stopniu ustrukturalizowania – jest miejsce na negocjację, wyrażanie zdolności przywódczych itp. Z tych powodów istotne znaczenie dla rozważań na temat wspomagania ma model decydenta, wokół którego niejako tworzą się swoiste warstwy „sił i środków" wspomagających.

Niegdyś pisałem [15], że podobnie jak za czasów Miltiadesa, zwycięzcy spod Maratonu, Hannibala, czy Napoleona, także i dziś do człowieka należy wybór czasu, miejsca i sposobu rozegrania bitwy. Człowiek ze swą intuicją, doświadczeniem, zdolnością myślenia twórczego jest niezastąpiony. Jednakże, aby te szczególne jego cechy mogły być w pełni wykorzystane w organizacjach, a zwłaszcza w procesach kierowania nimi, musi być wspomagany przez systemy informatyczne. Naturalna inteligencja musi być wspomagana przez inteligencję sztuczną. I w tym należy dostrzegać rolę i znaczenie systemów ekspertowych w zarządzaniu.

Systemy ekspertowe

Spory i kontrowersje wokół sztucznej inteligencji wydają się dobiegać kresu, a to za sprawą widocznego postępu w dziedzinie inżynierii wiedzy. Nie oznacza to, że przyczyny kontrowersji filozoficznej natury przestały być aktualne. Sztuczna inteligencja nie może uniknąć filozofii, jak zauważył J. McCarthy (1988).

Dla racjonalistycznej tradycji filozoficznej jest bliskie podejście H. A. Simona, zgodnie z którym istotą inteligencji tak sztucznej, jak i naturalnej, jest zdolność formalnego manipulowania symbolami reprezentującymi różne aspekty rzeczywistości. Można zatem mówić o paradygmacie formalno-symbolicznym w dziedzinie sztucznej inteligencji. Za alternatywne podejście można uznać paradygmat dystrybutywno-konektywistyczny.

Ogólny problem wspomagania działania, w szczególności dokonywania wyborów i podejmowania decyzji, bardzo często wiąże się z zagadnieniem tzw. zdrowego rozsądku. Aspektem wiedzy potocznej (zdroworozsądkowej), który nastręcza najwięcej bodaj trudności, jest jej kontekstowość (ang. *contextuality*). Próby z takimi metodami reprezentowania wiedzy zdroworozsądkowej, jak np. ramy (ang. *frames*), zbiory reguł w systemach produkcyjnych (ang. *production systems*) czy scenariusze (ang. *scripts*), wydają się niewystarczające.

Przypomnijmy, że sztuczna inteligencja, według E. Feigenbauma i P. McCorducka, jest:

> *...dziedziną informatyki (ang. computer science) związaną z koncepcjami i metodami wnioskowania symbolicznego, wykonywanego przez komputer, oraz z symboliczną reprezentacją wiedzy, używaną przy wnioskowaniu. Badania w dziedzinie sztucznej inteligencji mają zadanie dostarczyć komputerom zdolności, które ludzie uznają za inteligencję.*

Spośród licznych koncepcji rozwijanych w dziedzinie sztucznej inteligencji zwraca uwagę, klasyczna już koncepcja H. A. Simona, zakładająca analogię między funkcjonowaniem człowieka i komputera. Jej rozwinięciem jest teoria rozwiązywania problemów Newella i Simona, oparta na następujących założeniach:

1. Kilka i tylko kilka ogólnych właściwości systemu przetwarzania informacji przez człowieka jest niezmiennych w odniesieniu do każdego zadania i rozwiązującego problem.
2. Na podstawie tych właściwości można stwierdzić, że zadanie jest reprezentowane (w systemie przetwarzania informacji) jako

przestrzeń problemową oraz, że rozwiązanie problemu zachodzi w ramach tej przestrzeni.
3. Struktura zadania określa możliwe struktury przestrzeni problemowej.
4. Struktura przestrzeni problemowej określa możliwe strategie, jakie mogą być użyte do rozwiązania problemu.

Podstawą, niezmienną i ogólną właściwością każdego zadania jest to, że ma ono swoistą dla siebie strukturę, która dopuszcza lub uniemożliwia stosowanie takich, a nie innych operacji w celu jego rozwiązania.

System przetwarzania informacji (wg Simona i Newella) jest symboliczną, przestrzenną i dynamiczną reprezentacją struktury organizmu ludzkiego (a zwłaszcza centralnego układu nerwowego) oraz psychicznych funkcji, jakie na bazie tej struktury zachodzą.

Zadaniem określa się pewien stan rzeczy, pozostający w określonej relacji do podmiotu, który na tenże podmiot wpływa w sposób mobilizujący do jego podjęcia i rozwiązania. Na zadanie składają się zestaw informacji, obwarowanych szeregiem ograniczeń odnoszących się do ich przetwarzania, oraz informacje żądane, które stanowią efekt procesu przetwarzania informacji. W zadaniu niewiadomą jest sekwencja procesów (czynności), jakie trzeba wykonać, aby zadanie to rozwiązać.

Przestrzeń problemowa jest symboliczną indywidualną dla każdego systemu przetwarzania informacji, strukturą reprezentującą zadanie. Przyjęto sześć źródeł informacji warunkujących konstrukcję przestrzeni problemowej w konkretnym systemie przetwarzania informacji, a mianowicie:

- instrukcja żądania, która opisuje elementy środowiska zadaniowego oraz stanowi swoistą pamięć zewnętrzną,
- uprzednie doświadczenia z analogicznym zadaniem lub jego fragmentami,
- uprzednie doświadczenia z takimi samymi zadaniami,
- ogólne programy rozwiązania wielu różnych problemów,
- programy służące kombinowaniu danych z instrukcji zadania z innymi informacjami w celu konstrukcji przestrzeni problemowej i programów rozwiązania problemów w konkretnej sytuacji,
- informacje napływające w procesie rozwiązywania problemu, które mogą wpłynąć na zmianę przestrzeni problemowej.

Strategia jest ciągiem elementarnych lub ogólnych procesów, które muszą być wykonane, aby zadanie zostało rozwiązane. Zgodnie z koncepcją systemu przetwarzania informacji, strategia jest sposobem przeszukiwania przestrzeni problemowe przez procesor.

Aczkolwiek teoria przetwarzania informacji Simona i Newella jest uważana za propozycję klasyczną i nie pozbawioną różnych ograniczeń, to stanowi dobry punkt wyjścia do rozważań dotyczących systemów wspomagania zarządzania (decyzji), w tym systemów ekspertowych. Przyjmując rozróżnienie między **reprezentacją** (ang. *patterns*), tworzoną za pomocą symboli, oraz **środkiem** (ang. *medium*), użytym do jego implementacji, można stwierdzić, że jeżeli inteligencja pochodzi tylko ze struktur systemu symboli, to każdy środek, który dobrze implementuje poprawne wzorce i procesy, może osiągnąć inteligencję bez względu na to, czy jest wykonany z neuronów, obwodów logicznych, czy innych elementów.

Ekspertem *określa się specjalistę, rzeczoznawcę, który na podstawie swojej działalności lub wykorzystując czyjeś w tym zakresie osiągnięcia, bada konkretną rzeczywistość po to, by móc w sposób uzasadniony sformułować swoje oceny i wskazania.*

Do podstawowych rezultatów działalności ekspertów zalicza się:
- diagnozy i oceny,
- dyrektywy korekcyjne,
- dyrektywy optymalizujące (rozwojowe).

Systemem ekspertowym *określa się system informatyczny wykorzystujący wiedzę ekspertów w konkretnej dziedzinie dla zapewnienia wysokiej efektywności rozwiązywania zadań (sytuacji problemowych).*

Tak rozumiany system ekspertowy tworzą najczęściej następujące podsystemy:

- **baza wiedzy** zawierająca wymagane informacje o zadaniu (sytuacji problemowej),
- **baza danych** stanowiąca roboczą część systemu,
- **interpretator** wraz z **mechanizmami wnioskowania**,
- **procesor lingwinistyczny**, zapewniający dialogowe współdziałanie z użytkownikiem (np. decydentem) i/lub ekspertem.

Wiedza stanowi informacje, których potrzebuje system informatyczny, aby jego działanie można było uważać za inteligentne. Informacje te mogą mieć postać faktów i/lub reguł. Często wyróżnia się dwie kategorie wiedzy: „JAK" (ang. *knowledge „how"*) oraz „ŻE" (ang. *knowledge „that"*). Reprezentacja wiedzy jest swoistym systemem formalnym zapisu informacji posiadanych przez system wraz ze sposobami operowania na zbiorze tych informacji.

System reprezentacji wiedzy składa się z:

- notacji,
- semantyki notacji dla określonej dziedziny wiedzy,
- algorytmów wyszukiwania, wnioskowania i aktualizacji.

Bazę wiedzy tworzy wiedza dziedzinowa oraz reguły operacji na niej.

Formalnie **problem** w systemie ekspertowym definiuje się za pomocą trójki: (α, β, γ), gdzie:
α – kwalifikator problemu (np. CZY, KTÓRY, DLACZEGO);
β – specyfikator problemu, czyli zadanie orzekające (np. w problemach typu CZY, DLACZEGO) lub predykat (np. w problemach typu KTÓRY), określający co jest niewiadome i co powinno być rozstrzygnięte w procesie rozwiązywania problemu;
γ – założenia, czyli układ zadań orzekających fakty relewantne (bezpośrednio związane z problemem).

W zależności od charakteru związków przyczynowo-skutkowych między β i γ, problemy dzielą się na dwie klasy:

- **dedukacyjne** – gdy między γ i β zachodzi relacja przyczyny do skutku,
- **redukcyjne** – w przypadku odwrotnym. Odpowiada to dwom zasadniczym strategiom rozwiązywania problemów: strategii redukcyjnej oraz strategii dedukcyjnej.

Najczęściej przyjmuje się następujący podział narzędzi do budowy systemów ekspertowych:
- języki programowania:

a) zorientowane problemowo,
b) symboliczne,
- języki inżynierii wiedzy (ang. *knowledge engineering languages*):

a) systemy szkieletowe (ang. *skeletat systems*), np. Expert, KAS,
b) systemy hybrydowe (ang. *general purpose systems*), np. KEE, ART, OPS5,
- systemy wspomagające budowę systemów ekspertowych (ang. *system-building aids*),

a) środki wspomagające projektowanie (ang. *design aids*), np. AGE, TIMM,
b) środki wspomagające pozyskiwanie wiedzy (ang. *knowledge-aquisition aids*), np. Teiresias, Roget.

Wybór narzędzi do budowy systemu ekspertowego w zarządzaniu zależy od wielu czynników, w tym np. od: własności użytkowych narzędzia (sposób reprezentowania faktów i wiedzy niepewnej, sposób reprezentowania związków między faktami, możliwości strukturalizacji wiedzy, dostępne tryby wnioskowania), możliwości i prostoty obsługi, szybkości działania itp.

Wspomaganie działania

Przyjęto, że **działania** dowolne określa następująca struktura formalna:

$$/A = \langle A, \mathbb{A}, S, R, T, w, V \rangle$$

gdzie:
A – skończony zbiór działań elementarnych,
$\mathbb{A} = 2^A$ – zbiór wszystkich ciągów utworzonych z elementów zbioru A, zwanych **sekwencjami**,
$S = \{s = \sigma(u) : u \in A\}$ – zbiór bodźców, będących przyczyną sekwencji u,
T – zbiór chwil czasu,
w – relacja „wykonalności", określona na zbiorze $\mathbb{A} * R * T$, czyli $\langle u, r, t \rangle \in$ w określa, że wykonanie sekwencji $u \in A$ powoduje skutek $r \in R$ w chwili $t \in T$,
V – relacja „waluacji", określana na zbiorze $R * R$, czyli $\langle r_i, r_j \rangle \in V$, jeżeli skutek r_i jest wyżej oceniany („wartościowszy") niż r_j.

Systemem działania określa się następujący obiekt formalny:

$$AS = \langle /A, M, \alpha, \beta \rangle$$

gdzie:
M – zbiór realizatorów działania,
α – relacja możliwości działania, określona na zbiorze $M * A * T$, czyli $\langle m, a, t \rangle \in \alpha$ określa, że realizator $m \in M$ podejmuje działanie $a \in A$ w chwili $t \in T$,
β – relacja współdziałania określona na zbiorze $M * A * M$, czyli $\langle m, a, t \rangle \in \beta$ określa, że realizator $m \in M$ pozostaje w relacji (stosunku, interakcji) z innym realizatorem $n \in M$ ze względu na działanie $a \in A$.

Ponadto przyjęto, że z każdym systemem działania jest związany tzw. **kontekst** działania, czyli pewna klasa wszystkich kontekstów określających:
- warunki, w których działanie może być zainicjowane, a w szczególności:

– fizyczną wykonalność działania,
– psychologiczną dopuszczalność dla realizatora danego działania,
– społeczną dopuszczalność działania (zgodnie z rolą społeczną realizatora),
– organizacyjną dopuszczalność działania (np. akceptowalność w danej organizacji ze względu na zgodność z jej regulaminem, zwyczajami itp.),
- warunki, w których może być spowodowany określony skutek działania, np.:

– wystąpienie skutku, który chce się osiągnąć,
– nie wystąpienie skutku, którego chce się uniknąć,
– wykonanie działania, które powinno się wykonać,
– nie wykonanie działania, którego nie powinno się wykonywać itp.,
- warunki technicznej wykonalności działania, czyli istnienie niepustego zbioru metod (sposobów) działania oraz określonej liczby środków działania.

Systemem wspomagania działania będziemy nazywać kompleks specyficznych środków, zapewniających efektywne wspomaganie konkretnego działania w określonym systemie działania.

Dowolny system wspomagania można przedstawić w następującej postaci:

$$ASS = \langle I_A, P_A, M_A, D_A, W_{AC}, R_{AC}, M(A, C) \rangle$$

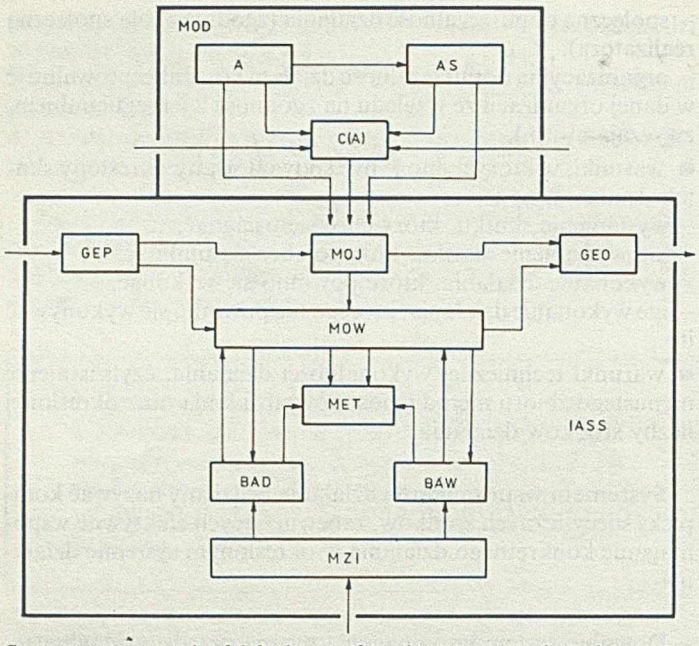
gdzie:
I_A – język działania (opisu działania, systemu działania, kontekstu, wspomagania),
P_A – zbiór możliwych pytań (dopuszczalnych w sensie J_A), dotyczących działania oraz jego kontekstu,
M_A – zbiór dopuszczalnych metod poszukiwania odpowiedzi na pytania (rozwiązywania zadań zawartych w pytaniach),
D_A – zbiór danych cech i własności działania i systemu działania,
W_{AC} – zbiór wiedzy o działaniu i jego kontekście (fakty, zdarzenia, reguły),
R_{AC} – zbiór reguł wnioskowania, czyli formułowania logicznie poprawnych odpowiedzi na pytania z P_A na podstawie D_A i W_{AC},
$M(A, C)$ – model działania i kontekstu działania (lub model decydenta).

Jeżeli ASS pozbawimy W_{AC} i R_{AC}, to otrzymujemy „klasyczny" system wspomagania (np. typu DSS, CAD, CAL, itp.). W przeciwnym wypadku mamy do czynienia z systemem wspomagania, którego istotną cechą są elementy sztucznej inteligencji (np. systemu typu Expert lub Expert DSS).

Tak rozumiany system wspomagania działania (ASS) można przedstawić w postaci pewnej struktury modułowej (rys.), zawierającej następujące moduły:

MODUŁ MODELI (MOD),
MODUŁ METOD (MET),
GENERATOR PYTAŃ (GEP),
GENERATOR ODPOWIEDZI (GEO),
MODUŁ JĘZYKA (MOJ),
MODUŁ WNIOSKOWANIA (MOW),
BAZĘ DANYCH (BAD),
BAZĘ WIEDZY (BAW),
MODUŁ ZDOBYWANIA INFORMACJI (DANYCH I WIEDZY) (MZI).

Każdy z modułów stanowi z jednej strony złożoną strukturę formalną (logiczną), z drugiej zaś – strukturę techniczno-programową.



System wspomagania działania przedstawiony w postaci struktury modułowej

Informatycznym systemem wspomagania działania (IASS) określać będziemy taki system informatyczny, który zapewnia efektywną, komputerową realizację wspomagania określonego działania.

Do podstawowych funkcji IASS należy zaliczyć funkcje:
- tworzenia modelu działania,
- tworzenia struktur danych,
- reprezentacji wiedzy, interpretacji wiedzy,
- rozwiązywania zadań na modelach obliczeniowych,
- poszukiwanie odpowiedzi na modelach wiedzy,
- wnioskowania na podstawie reguł logiki (klasycznej, modalnej, wielowartościowej, rozmytej),
- formowania pytań,
- formowania odpowiedzi,
- aktualizacji danych i wiedzy,
- restrukturyzacji baz danych i baz wiedzy itp.

Łatwo dostrzec, że warunki projektowania systemu typu DSS (ang. *decision support system*) na bazie IASS są oczywiste. Model AS jest formułowany w kategoriach DS (*decision system* lub *decision situations*), zaś moduł metod redukuje się do metod podejmowania decyzji i dokonywania wyborów.

Modele decydenta

Pomińmy w tym miejscu klasyczne modele, znane nader dobrze z teorii decyzji i badań operacyjnych, aczkolwiek od nich właśnie należałoby rozpocząć uwagi na temat modeli decyzyjnych w zarządzaniu. Na gruncie psychologii w klasie modeli poznawczych, dzięki teorii decyzji, jak zauważa J. Kozielecki [7], są tworzone operacyjne modele człowieka. W przeciwieństwie do behawiorystów, traktujących jednostkę jako system reaktywny, oraz psychoanalityków, którzy lansowali koncepcję człowieka niedoskonałego, model operacyjny jest modelem człowieka-organizatora, który aktywnie osiąga określone (pożądane) wartości.

Wyrazem tej aktywności jest rozwiązywanie zadań, tj. sytuacji, w której istnieje rozbieżność między stanem rzeczywistym $\underline{x}$ o wartości $v(\underline{x})$, a stanem antycypowanym o wartości $v(\bar{x})$, gdzie $v(\underline{x})$ jest aktualnym stanem posiadanych wartości, zaś wartość $v(\bar{x})$ jest standardem zachowania. W przypadku powstania rozbieżności: $\Delta(\underline{x}, \bar{x}) = |v(\bar{x}) - v(\underline{x})| \neq 0$ rodzi się tzw. napięcie motywacyjne, powodujące, iż człowiek podejmuje działania ukierunkowane na redukcję deficytu wartości. Gdy $\Delta = 0$, człowiek podejmuje działania ukierunkowane na zachowanie status quo. Występuje tu motywacja homeostatyczna, zwana także motywacją braku (deficytu).

Bardziej złożonym jest mechanizm motywacji heterostatycznej, zwany również motywacją wzrostu. Zgodnie z nią, osiągnięcie określonych wartości nie tyle redukuje napięcie, ile je utrzymuje, a nawet powiększa. Człowiek ma pewien poziom aspiracji v*, najczęściej o zmiennej strukturze. W przypadku powstania rozbieżności $\Delta(v^*, v(\underline{x})) \neq 0$ rodzi się napięcie aspiracyjne i następuje poszukiwanie nowych wartości. Zdobycie ich nie niweluje Δ, lecz często je utrzymuje lub powiększa. Zapewne człowiek jest systemem zbyt złożonym, aby go było można opisać tylko w kategoriach.

Podejmowanie decyzji polega także na integrowaniu i selekcji informacji, zwłaszcza w razie przeddecyzyjnej. Szczególną rolę należy przypisać danym pozwalającym określić podstawowe wymiary:
- pozytywną wartość sukcesu $v(s)$,
- negatywną z reguły wartość ryzyka $v(x)$,
- wartość autoteliczną samego działania $v(x)$.

Ludzie różnią się między innymi i tym, że tym wymiarom przypisują różne wagi. Z kolei, na zwiększenie napięcia motywacyjnego i trwałość czynności decyzyjnych mają wpływ emocje antycypowane (np. przewidywana duma po odniesieniu sukcesu lub poczucie winy po niepowodzeniu) oraz aktualne (przeżywanie: radość lub lęk). Z psychologicznego punktu widzenia jest oczywiste, że motywacja pełni trzy podstawowe funkcje w życiu człowieka: uruchamia działania celowe, nadaje im kierunek oraz podtrzymuje je w zmieniających się okolicznościach.

Trzy podstawowe grupy czynników tworzą względnie pełny wizerunek „homo decernens", a mianowicie:
- cechy osobowości (motywacja, poziom charakteru, jego dynamizm i szerokość),
- cechy otoczenia (środowiska) wpływające na zachowania decyzyjne człowieka (np. tzw. klimat organizacyjny),
- cechy działania (np. jego struktura rzeczowa, przestrzenna, czasowa).

Sytuację decyzyjną można więc określić jako zbiór relacji: (stosunków) określony na produkcie kartezjańskim zbioru cech decydenta (osobowość), otoczenia (środowiska) i działania (proces). Analiza systemowa różnorodnych sytuacji decyzyjnych należy obecnie do najważniejszych i najpilniejszych problemów poznawczych i praktycznych [3].

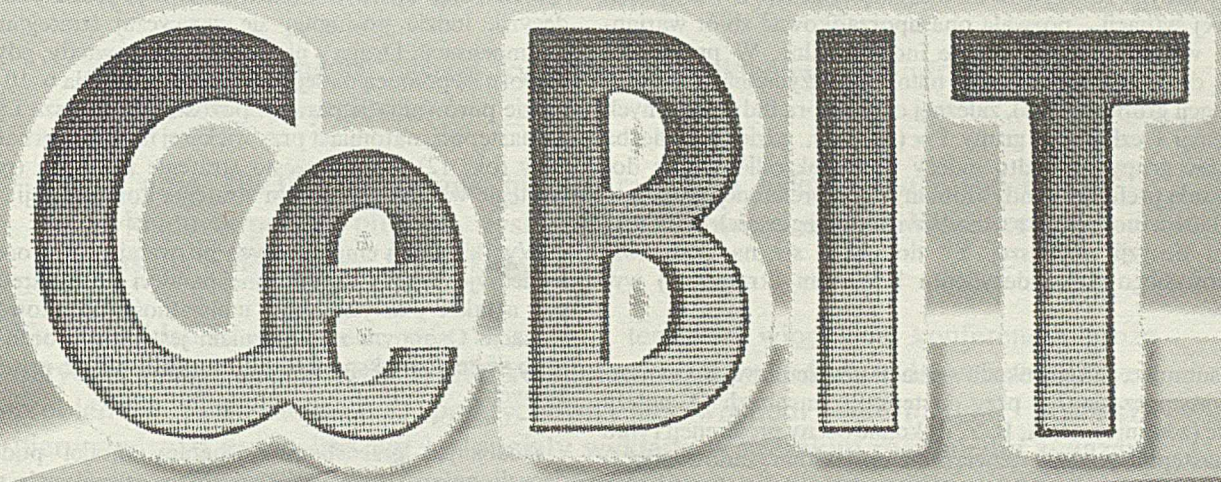
Na szczególną uwagę zasługują następujące modele decyzji:
- neoklasyczny model racjonalności decyzji, oparty na maksymalizacji funkcji użyteczności,
- behawiorystyczny model decyzji zadowalających, operujący poziomami aspiracji i satysfakcji (H. A. Simon),
- model racjonalności ewolucyjnej, obejmujący dylematy społeczne i tzw. pułapki społeczne (A. Rapaport),
- model holistycznej racjonalności ekspertów (R. Dreyfus),
- model racjonalnego planowania programowo-celowego (G. S. Pospiełow),
- model racjonalności quasi-zadowalającej, wykorzystujący tzw. funkcje osiągnięcia, związany z problematyką DSS (A. Wierzbicki).

Szczegółowa charakterystyka każdego z wymienionych modeli przekracza ramy niniejszego artykułu. →

Imponujący światowy pokaz techniki biurowej, informatycznej i telekomunikacyjnej...

Zdobycie pełnej informacji w jednym miejscu to marzenie nowoczesnego biznesmena, bo przecież time is money! Tę unikalną, jedyną na świecie szansę kompleksowego przeglądu wszystkich, zgromadzonych w jednym miejscu, dziedzin techniki biurowej, informatycznej i telekomunikacyjnej dają największe i najważniejsze dla branży targi międzynarodowe z udziałem 6000 wystawców z ponad 45 krajów. Całościowa światowa oferta najnowszych osiągnięć podana na tacy. Tylko na CeBIT w

Hanowerze 24.– 31. 03. 1993



Światowe Centrum • Biuro • Informacja • Telekomunikacja

Bliższe informacje: MGR ZDZISŁAW A. GEPPERT, PRZEDSTAWICIEL W POLSCE
ul. Mokotowska 49, 00-512 Warszawa
Tel.: (02) 66 00-505, 66 00-525, Telex: 814241 mex pl, Fax: (022) 29 85 59

DEUTSCHE MESSE AG, HANNOVER/GERMANY

65192

Przedstawimy teraz formalny model sytuacji decyzyjnej dla przypadku indywidualnego decydenta [8]:

$\langle So, T, Q / S, A, B, Y, f, K, Y^* \rangle$

gdzie:
So – początkowa sytuacja (opisywana najczęściej werbalnie w postaci zadania),
T – czas na podjęcie decyzji,
Q – zasoby niezbędne na podjęcie decyzji,
$S = (S_i ..... S_n)$ – zbiór sytuacji alternatywnych, dodatkowo określających sytuację So,
$A = (A_i ....., A_k)$ – zbiór celów – osiąganych w rezultacie rozwiązania zadania,
$B = (B_i ....., B_l)$ – zbiór ograniczeń,
$F = (Y_i ....., Y_m)$ – zbiór wariantów rozwiązań alternatywnych,
f – funkcja preferencji decydenta,
K – kryterium (kryteria) wyboru najlepszego rozwiązania,
Y* – rozwiązanie optymalne.

Czynniki (So, T, Q) najczęściej są znane, jednakże zdarza się, że muszą być określone przez decydenta. Do zasobów Q zalicza się np. wiedzę i doświadczenie decydenta oraz ekspertów, potencjał naukowo-badawczy określonych instytucji (instytutów naukowych, instytutów doradczo-eksperckich), systemy informatyczne itp. Zbiór S tworzą sytuacje hipotetyczne (hipotezy, wersje, scenariusze), którym są przypisywane wartości subiektywnych prawdopodobieństw.

Dla poszczególnych celów ze zbioru A należy określić kryteria ich osiągnięcia, wskaźniki stopnia osiągnięcia celów oraz ich priorytety. Decydowanie odbywa się zawsze w warunkach różnorodnych ograniczeń (prawnych, finansowych, materiałowych, kadrowych itp.) i dlatego należy precyzyjnie zdefiniować zbiór ograniczeń B, zawężający niejako obszar poszukiwań rozwiązania zadania. Aby osiągnąć zbiór celów A jest definiowany, a raczej generowany, zbiór alternatywnych wariantów decyzji – rozwiązania zadania Y, z którego należy wybrać decyzję optymalną Y* (lub zadowalającą, gdy zdefiniowany będzie poziom satysfakcji). Funkcja preferencji f jest wykorzystywana do opisu wyrażenia – jakościowego lub ilościowego – oceny decyzji ze względu na osiągnięcia celów w warunkach możliwej sytuacji – pozwala ona uporządkować zbiór wariantów ze względu na preferencje indywidualne. W przypadku modeli decydenta grupowego mamy do czynienia z funkcją preferencji grupowej F (f), zależnej od wektora indywidualnych preferencji i członków grupy $f = (f_1 ..., f_d)$, gdzie: d – liczba członków grupy. Ponadto należy wtedy określić zasadę dostosowania preferencji indywidualnych do preferencji grupowej (przykładem może być np. zasada większości głosów). Wreszcie, wyboru decyzji najlepszej Y* dokonuje się na podstawie zdefiniowanego przez decydenta kryterium (kryteriów) wyboru K.

Reasumując, w warunkach sytuacji problemowej So, w dyspozycyjnym czasie T i przy dostępnych zasobach Q, należy określić (zdefiniować) A, B, Y, dokonać oceny preferencji f (lub F), a następnie znaleźć rozwiązanie optymalne (zadowalające) $Y^* \subset Y$, posługując się w tym celu kryterium (kryteriami) wyboru K.

Dla podejmowania decyzji indywidualnych lub grupowych charakterystyczne są następujące cechy:
- wiele elementów sytuacji (zadania) ma przede wszystkim werbalny charakter, niektóre tylko częściowo można wyrazić w sposób ilościowy, a niektóre pozostają wręcz nieznane;
- pełne określenie nieznanych elementów, z których część dotyczy wszak przyszłości, jest niemożliwe za pomocą metod algorytmicznych, a zatem intuicja i doświadczenie (także emocje) mają znaczący wpływ na strukturyzację zadania oraz poszukiwania jego rozwiązania:
- część charakterystyk zadania można „zmierzyć" lub oszacować posługując się obiektywnymi metodami i technikami analitycznymi (np. statystycznymi), lecz wiele istotnych elementów można „zmierzyć" subiektywnie (np. priorytety celów, preferencje decydentów);
- rozwiązanie sytuacji decyzyjnych odbywa się najczęściej w warunkach nieokreśloności, wynikających z niepełnego opisu zadania oraz pewnego przewidzenia przyszłych warunków działania, a więc i oczekiwanych wyników (korzyści, strat);
- nie można wykluczyć takich sytuacji, w których podjęte decyzje mogą bezpośrednio naruszać interesy decydentów lub ekspertów, a więc ich motywacja wpływa na wybory rozwiązań.

Analizując wiele różnych sytuacji decyzyjnych można wyodrębnić pewne operacje elementarne, do których można zaliczyć:
- operacje z kryteriami:
  - wyodrębnienie kryteriów,
  - wyznaczanie wag kryteriów,
  - uporządkowanie kryteriów,
- operacje z alternatywami według kryteriów:
  - porównanie dwóch ocen alternatyw według skali jednego kryterium,
  - porównanie zmian ocen dwóch kryteriów,
  - ilościowe określenie zmiany oceny jednego kryterium w zależności od zmiany drugiego kryterium,
  - określenie zadowalającej wartości jednego kryterium,
- operacje z alternatywami:
  - porównanie dwóch alternatywnych wariantów i wyróżnienie lepszego z nich,
- wyróżnienie wśród kryteriów tych, których wartość może ulec zmianie „bez szkody dla zadania", tj. tych, których wartość można zmniejszyć lub zwiększyć,
  - ocena wariantów wg kryteriów niezadowalających w maksymalnym stopniu,
- operacje wyboru:
  - wybór lepszego wariantu z grupy (zestawu, zbioru),
  - poszukiwanie „idealnej" alternatywy.

Realizując operacje elementarne w procesie podejmowania decyzji, ludzie posługują się zazwyczaj strategiami selekcji i kompensacji. Drugą z nich – najczęściej wtedy, gdy dokonują wyboru za pomocą małej liczby kryteriów. Dla 6–10 alternatyw ludzie porównują je parami, pozostawiają lepszą i przechodzą do następnej, natomiast przy większej liczbie alternatyw i kryteriów (ok. 12) stosują często strategie mieszane (początkowo strategie wyłączenia, potem strategie kompensacji).

Wyniki badań empirycznych pozwalają także ocenić wpływ na decyzje wyboru ograniczonej pamięci o bezpośrednim dostępie, a także plastyczności i umiejętności adaptowania się do sytuacji. Osobnym zagadnieniem jest wpływ ocen ekspertów i/lub systemów ekspertowych na efektywność procesów decyzyjnych.

Analizując przedstawiony ogólny model podejmowania decyzji należy stwierdzić, że obszar wspólnych problemów poznawczych i praktycznych teorii decyzji i sztucznej inteligencji jest bardzo liczny, zaś w wielu przypadkach zbieżność dociekań jest całkowita. A to oznacza także, że dziś wizji „homo decernens" nie można z powodzeniem kreować bez sięgania do osiągnięć w dziedzinie sztucznej inteligencji. Z kolei, postęp w sztucznej inteligencji byłby zapewne mniejszy, gdyby nie nowa wiedza na temat zachowań decyzyjnych człowieka, uzyskana m.in. dzięki teorii decyzji oraz psychologii poznawczej.

dokończenie na s. 21

Call for papers

Piąta Jubileuszowa Wyższa Międzynarodowa Górska Szkoła PTI

Szkoła odbędzie się w Szczyrku w dniach od 21 do 25 czerwca 1993. W szkole weźmie prawdopodobnie udział około stu pięćdziesięciu osób, przedstawicieli wszystkich dziedzin informatyki: pracowników uczelni, programistów, szefów ośrodków obliczeniowych i przedstawicieli kierownictw polskich i zagranicznych firm informatycznych. Podobnie jak w latach poprzednich oprócz przedstawienia praktycznych osiągnięć polskich informatyków Szkoła będzie miała swój temat wiodący. Chcemy aby w tym roku były nim problemy programowania obiektowego, sterowanego zdarzeniami, czy jak jeszcze można nazwać tę względnie nową filozofię tworzenia programów. Serdecznie zapraszamy do nadsyłania propozycji treści wykładów.

W trakcie trwania Szkoły przewidujemy wzorem lat ubiegłych organizację równoległych sesji o charakterze marketingowym. Będą one okazją dla producentów rozwiązań informatycznych do zaprezentowania ich szerokiemu gronu potencjalnych użytkowników.

Prosimy osoby zainteresowane prezentacją wykładów w trakcie Szkoły o dotrzymanie następujących terminów, ze swojej strony zobowiązując się do dotrzymania terminu zależnego od nas:

do dnia 1993.03.15 nadesłanie zgłoszenia wykładu zawierające: tytuł wykładu, nazwisko Autora, ewentualną nazwę instytucji, którą Autor reprezentuje oraz liczący do stu słów konspekt wykładu.

do dnia 1993.04.01 nastąpi kwalifikacja wykładów i poinformowanie Autorów o jej wynikach.

do dnia 1993.05.15 (termin nieprzekraczalny) należy dostarczyć treść wykładu w postaci pliku na dyskietce (preferowany edytor: WordPerfect, prosimy o dołączenie wydruku).

Treść wykładów nie musi być związana z tematem wiodącym konferencji, może dotyczyć również problemów grafiki komputerowej, systemów baz danych oraz innych problemów związanych z praktycznym uprawianiem informatyki. Jednakże pierwszeństwo w kwalifikacji będą miały wykłady w szerokim rozumieniu związane z tematem wiodącym. Zgłoszenia wykładów oraz propozycje prezentacji marketingowych prosimy nadsyłać na adres:

POLSKIE TOWARZYSTWO INFORMATYCZNE
ODDZIAŁ GÓRNOŚLĄSKI
40-014 Katowice
ul. Mariacka 6
tel/fax. 538-102

JERZY KISIELNICKI
Warszawa

Informacyjna infrastruktura zarządzania – potrzeby i tendencje zmian

Informacyjna infrastruktura zarządzania stanowi element wspomagający lub opóźniający podejmowanie decyzji na różnych szczeblach decyzyjnych. Historia najnowszego okresu uczy, że w naszym kraju najczęściej był to element właśnie opóźniający. W systemie centralnego planowania, nazywanego również nakazowo-rozdzielczym lub administracyjno-nakazowym, rola informacyjnej infrastruktury zarządzania była o wiele mniejsza niż w systemie gospodarki rynkowej. Jak pisze Alvin Toffler w książce „Power Shift" cały świat czeka totalna informacyjna wojna, która przebiegać będzie przez całą światową gospodarkę [5].

Każdy system zarządzania, jeżeli ma być efektywny, powinien być wspierany przez odpowiednią infrastrukturę zarządzania. We współczesnych warunkach jest ona oparta na technice komputerowej. Wydzielenie informacyjnej infrastruktury zarządzania (IIZ) z systemu zarządzania jest konstrukcją teoretyczną, ponieważ właśnie ona jest jednym z elementów systemu zarządzania.

Rozpatrując IIZ należy odpowiedzieć na dwa pytania:
1. Jaki jest obecny stan tej infrastruktury?
2. Jakie istnieją tendencje jej zmiany?

Informacyjna infrastruktura zarządzania IIZ jest określona jako następująca uporządkowana czwórka symboli *I, M, K, R,* [2]:
I – zasoby informacyjne,
M – modele,
K – środki techniczne wspomagające system zarządzania,
R – relacje.

Odziedziczona po centralnym planowaniu infrastruktura jest, jak już wspomniano, nieprzydatna dla nowego systemu zarządzania. Była ona dostosowywana do wieloszczeblowego hierarchicznego systemu podejmowania decyzji. System ten był zbudowany głównie dla potrzeb centralnych organów władzy i to zarówno gospodarczej, jak i politycznej. Przedsiębiorstwa i inne podstawowe jednostki organizacyjne były w zasadzie tylko dostawcami informacji.

Jak wielkie były to różnice, pokazano w tabeli oraz na przykładzie różnicy w charakterystyce modeli używanych w systemie wspomagania decyzji (SWD), tzw. modeli klasy DDS. Wychodzi się tu z założenia, że właśnie na tego typu systemach najdokładniej można prześledzić konieczność zmian w systemach informatycznych. System wspomagania decyzji w systemie centralnego zarządzania (systemie administracyjno-nakazowym) był oparty na modelu, który można scharakteryzować następująco:
- stosuje formalny język opisu,
- ma postać bilansową (zadaniem nadrzędnym jest zbilansowanie planu),
- dotyczy jednostkowych decyzji,
- zależności w modelu są przedstawione w postaci liniowej – stąd popularność wszystkich modeli związanych z programowaniem liniowym,
- jest budowany na określony termin,
- użyte w modelu informacje mają charakter deterministyczny,
- obliczenia odbywają się najczęściej za pomocą komputerowej trzeciej generacji (System Riad lub IBM 360 i 370).

Dla systemów wspomagania decyzji działających w warunkach gospodarki rynkowej są potrzebne zupełnie inne modele. Modele te możemy w uproszczeniu scharakteryzować następująco:
- mieszany język opisu, a więc obok opisu formalnego funkcjonują tzw. zmienne semantyczne,
- postać modelu można określić terminem postaci systemowej a więc takiej, którą stosuje się w badaniach systematycznych, i która jest zgodna z teorią systemów,
- może wspomagać równocześnie cały blok różnorodnych, niezależnych i zależnych decyzji,
- występujące w modelu zależności są powiązane w różnorodny sposób, a więc nieliniowo, stochastycznie, logicznie itp.,
- uwzględniają element czasu, czyli zaliczają się do dynamicznych,
- metody rozwiązania problemu są metodami zarówno symulacji komputerowej, jak i metodami heurystycznymi,
- w zależności od wielkości problemu modele są wyliczane na komputerach osobistych albo na komputerach pracujących w sieci.

Trudności, jakie wynikają przy stosowaniu tego typu systemów wynikają z faktu, że w porównaniu z poziomem światowym istniejąca w Polsce technika komputerowa była i jest przestarzała i to zarówno w kategorii sprzętu, jak i oprogramowania. Mimo, że w ostatnim okresie opóźnienie to znacznie zmniejszyło się, szczególnie w dziedzinie komputerów osobistych, to jednak nadal jest ono znaczne. Istotne jest tu zwłaszcza opóźnienie w zakresie sieci komputerowych, wielodostępnych i rozproszonych baz danych oraz dużych komputerów. Niewielki jest też dostęp do światowych baz wiedzy.

Najbardziej istotną przyczyną takiego stanu rzeczy jest brak środków finansowych. Odziedziczony po dawnym systemie stereotyp myślenia, to niedocenianie wagi wszelkiego typu informacji oraz technik jej przetwarzania i przesyłania.

Do analizy informacyjnej infrastruktury zarządzania posłużono się pojęciem generacji IIZ. Jest to pojęcie umowne i stanowi modyfikację i rozszerzenie pojęcia generacji komputerów. Kryteria podziału zostały określone na podstawie pojęcia IIZ. Podział na generacje przedstawiono w syntetyczny sposób w tab. 2. Tabela ta jest tak skonstruowana, że w górnej

Tabela 1. Morfologia głównych typów modeli wspomagających zarządzanie

L.p.	Kryterium konstrukcji modelu	Model - typ
1	Język opisu	opisowy (formalny) (mieszany)
2	Postać modelu	(bilansowa) optymalizacyjna ekonomiczna (systemowa)
3	Wielorakość decyzji	(jednostrukturalna) (wielostrukturalna)
4	Zapis funkcji kryterialnej	(jednokryterialne) (wielokryterialne) bezkryterialne
5	Sposób prezentacji zależności	dyskretne (liniowe) nieliniowe (mieszane)
6	Element czasu	(statyczny) (dynamiczny)
7	Metoda rozwiązania	(programowanie matematyczne) metody graficzne (symulacja) (heurystyki)
8	Charakterystyka informacji	(deterministyczna) (probabilistyczna) statystyczna strategiczna
9	Środki techniczne	komputery osobiste (średnie i duże komputery) (sieci komputerowe) bez komputera
10	Dziedzina	(inwestycje, zaopatrzenie, transport, kompleksowy zbyt)
11	Formuła zarządzania	(rynkowa) parametryczna (administracyjno-nakazowa)

źródło: opracowanie własne

( ) modele stosowane najczęściej w praktyce gospodarczej

( ) modele najbardziej potrzebne z punktu widzenia potrzeb gospodarki /konstrukcja modeli przyszłościowych/

części przedstawiono najbardziej istotne cechy kolejnych generacji IIZ. Dolna część tabeli pozwala odpowiedzieć na pytanie, czy do rozwiązania postawionego problemu dysponujemy IIZ odpowiedniej generacji. Może być również sytuacja, że generacja systemu jest za wysoka, a więc występuje nieekonomiczne wykorzystanie dysponowanych zasobów. Można wydzielić następujące cztery generacje IIZ.

I – systemy ewidencyjno-sprawozdawcze,
II – systemy informowania kierownictwa (MIS), wyposażone w bazę danych,
III – systemy wspomagania decyzji (DSS), wyposażone w bazę danych oraz w bazę modeli,
IV – systemy ekspertowe (ES), wyposażone zarówno w bazę danych i modeli, jak i bazę wiedzy.

Gospodarka działająca w systemie administracyjno-nakazowym, opierała się na systemach ewidencyjno-sprawozdawczych. W tego typu systemach zasoby informacyjne zawierały informacje o charakterze retrospektywnym i deterministycznym, a stosowne modele miały charakter równań bilansowych typu modeli przepływów gałęziowych Lontiewa.

W analizie prezentowanego problemu należy odpowiedzieć na pytania:
1. Jaka IIZ jest potrzebna Polsce do efektywnego zarządzania?..
2. Czy do rozwiązania naszych problemów istnieją za granicą odpowiedniej klasy systemy komputerowe?

W systemach fizycznych sytuacja wydaje się nieco prostsza. W 1985 r. David Deutsch z Uniwersytetu w Oxfordzie wykazał, że istnieje uniwersalny komputer kwantowy, który daje się tak zaprogramować, że wykonuje dowolne zadania wykonywane przez jakikolwiek obiekt fizyczny. Dla systemów społecznych taka sytuacja prawdopodobnie będzie możliwa do wykonania dopiero przy zastosowaniu neurokomputerów.

Dzieląc wg H. A. Simona [4] zadania w sferze zarządzania na dobrze, słabo oraz źle ustrukturalizowane możemy stwierdzić, że większość naszych problemów należy zaliczyć do klasy źle ustrukturalizowanych, a tylko niektóre do klasy ustrukturalizowanych słabo. Jak już wspomniano do rozwiązania naszych problemów dysponujemy określonymi czteroma generacjami systemów infrastruktury komputerowej.

Tabela 2. Podział generacji komputerowej infrastruktury

Cechy	Generacja			
	I	II	III	IV
Zasoby informacyjne	Deterministyczne, pełne, porównywalne, wiarygodne	Deterministyczne, nie zawsze pełne i porównywalne	Probabilistyczne, statystyczne	Strategiczne, pochodzące z różnorodnych źródeł
Modele	Proste, oparte na czterech działaniach, procenty	Proste modele matematyczne typu równania, przepływy międzygałęziowe	Optymalizacyjne o dowolnej złożoności, symulacyjne	Heurystyczne, logiczne
Środki techniczne podstawowe	Komputery wyposażone w duże pamięci masowe (taśmy i dyski magnetyczne), oprogramowanie ukierunkowane problemowo, języki COBOL PC-1, FORTRAN	Komputery pozwalające na konwersacyjny tryb pracy i pracujące w czasie rzeczywistym, odpowiednie oprogramowanie konwersacyjne, języki manipulacyjne i opisu danych	Komputery pracujące z bardzo dużą szybkością, wyposażone w koprocesory, systemy zarządzania pojęciem danych i modeli, języki wysokiego poziomu oraz specjalistyczne typu GPSS DYNAMO	Komputery o bardzo dużej mocy pracujące w sieciach, superkomputery o praktycznie nieograniczonej pamięci. Języki zbliżone do naturalnych. Systemy zarządzania bazą wiedzy
Środki techniczne uzupełniające	Urządzenia tzw. małej informatyki	Lokalne sieci danych	Sieci komputerowe homo- i heterogeniczne, poczta elektroniczna, skomputeryzowane biuro	Łączność satelitarna, inteligentne terminale
Problemy	Ewidencyjno-statystyczne Retrospektywne analizy szeregów czasowych Wielowskaźnikowe analizy ekonomiczne	Szybkie wyszukiwanie informacji Informowanie kierownictwa (tzw. monitory) Wspomaganie zarządzania operacyjnego	Wspomaganie decyzji taktycznych i strategicznych Sterowanie produkcją	Słabo i źle ustrukturalizowane wymagające udziału ekspertów.Realizacja zadań z zakresu sztucznej inteligencji

Źródło: opracowanie własne

Trudno jest precyzyjnie określić, kiedy zakończyła się jedna generacja, a rozpoczyna się następna. Można z pewnym uproszczeniem przyjąć, że pierwsza generacja to już historia. Pozostałe generacje nadal rozwijają się i doskonalą. Stale odbywa się rozwój określonych generacji, który możemy określić mianem rozwoju spiralnego. Generacje wzajemnie się przeplatają i uzupełniają. Współczesna IIZ pozwala między innymi na zmieniające się otoczenie. Tylko organizacja, w której funkcjonuje trzecia i czwarta generacja IIZ, a więc systemy klasy systemów wspomagania decyzji oraz systemów ekspertowych, zapewnia realizację takich zadań, jak badanie otoczenia organizacji oraz wybór efektywnej strategii działania. Jest to bowiem narzędzie, pozwalające na stały przepływ informacji między otoczeniem a organizacją.

Rozwój IIZ idzie również w tym kierunku, aby stale badać i analizować tzw. słabe sygnały nadchodzących zmian. Generacja zerowa czyli pochodząca z okresu przed komputeryzacją, jak również pierwsza generacja IIZ odbierały tylko tzw. silne sygnały. Podobnie można powiedzieć o niepełnych wyższych generacjach (niepełne generacje to takie, w których istnieją tylko wybrane elementy triady, a więc np. zasoby informacyjne i modele charakterystyczne dla generacji drugiej, natomiast środki techniczne – charakterystyczne dla generacji trzeciej).

Jeżeli IIZ odbiera tylko silne sygnały to najczęściej jest już zbyt późno, aby można było zmienić strategię działania lub przeprowadzić zmiany strukturalne. Oczywiście pomijamy sytuacje, kiedy takie sygnały odbierze nie zawsze najlepiej przygotowana kadra organizacji.

W gospodarce powinna istnieć taka IIZ, aby jak najwcześniej odbierać nawet słabe sygnały, o ile one mają lub będą miały wpływ na działalność organizacji, a więc również na jej wyniki. Mówimy więc, że współczesna IIZ pozwala realizować proces monitoringu gospodarczego jako proces odbierania i przekształcania wszelkiego tego rodzaju sygnałów. Sygnały takie mają różnorodną wartość i znaczenie, a więc dotyczyć mogą zarówno zagrożeń, jak i powstania pozytywnych warunków. Jak pisze N. T. Ansoff w pracy „Zarządzenie strategiczne" [4] w ciągu XX wieku historycznie rozwinięte umiejętności przedsiębiorstwa znajdowały coraz mniejsze zastosowanie do kluczowych zmian w otoczeniu. W rezultacie potrzebny na efektywną reakcję czas coraz bardziej wzrastał. Najnowsze generacje IIZ mają właśnie możliwości skracania czasu tej reakcji, o której pisze H. T. Ansoff. Nie można więc unowocześniać mechanizmów funkcjonowania gospodarki bez zmiany i unowocześnienia IIZ.

Zarówno trzecia, jak i czwarta generacja IIZ powodują, że współczesny decydent ma do dyspozycji bardzo silne narzędzia. Należy również pamiętać o powstających zrębach następnej generacji IIZ z tzw. sztuczną inteligencją, opartą na możliwościach technicznych neurokomputerów.

W konsekwencji decydent, jeżeli jest odpowiednio przygotowany, może realizować nawet najbardziej złożone zadania pod warunkiem, że zna IIZ i co ważniejsze, potrafi się nią posługiwać. Powinien też zdawać sobie sprawę z uwarunkowań w jej stosowaniu. Jest to jednak oddzielny problem, który należy tu tylko zasygnalizować.

Ponieważ w Polsce przeważają nadal systemy pierwszej generacji, a istniejące systemy drugiej generacji znajdują się dopiero w początkowym okresie rozwoju, to można przyjąć, że przyszłościowy kierunek działań w zakresie unowocześnienia systemu zarządzania oraz dogonienie, jak pisze S. Kwiatkowski, „uciekającego świata" [3], to powszechność stosowania systemów drugiej generacji z dużymi relacyjnymi bazami danych, a także systemów wspomagania decyzji i systemów ekspertowych.

LITERATURA

[1] Ansoff H. I.: Zarządzanie strategiczne. PWE, Warszawa 1985
[2] Kisielnicki J.: Informatyczna struktura zarządzania. PWN, Warszawa 1992
[3] Kwiatkowski S.: Uciekający świat. PWN, Warszawa 1991
[4] Simon H. A.: Podejmowanie decyzji kierowniczych. PWE, Warszawa 1992
[5] Toffer A.: Power Shift. Bantam Books, 1990

Mikroprocesory o podwójnej częstotliwości zegara

Do niedawna, wybierając nowy komputer typu PC, mieliśmy do dyspozycji bądź tradycyjny system oparty na mikroprocesorze 80286, bądź też szybszy, lecz droższy, oparty na 80386. Obecnie firma Intel Corp. oferuje dziesiątki różnych mikroprocesorów o bardzo subtelnych różnicach między poszczególnymi modelami.

Ostatnio wprowadzono technikę podwajania częstotliwości zegarowej, co jeszcze bardziej zaciemnia ten obraz. Nie wymaga to od producentów przyspieszania częstotliwości układów na płycie głównej, ponieważ sprzedawcy PC po prostu wstawiają kostkę zegara o częstotliwości np. 50 MHz do płyty, stosowanej poprzednio przy częstotliwości 25 MHz.

Pierwszym takim procesorem, podwajającym częstotliwość zegara, był układ i486 DX2, który pojawił się w sprzedaży na początku 1992 r. Może on realizować wszystkie funkcje procesora i486 DX znacznie szybciej. Kostka i486 DX2 zawiera układ zarządzania pamięcią, układ zmiennego przecinka oraz osiem kbajtów pamięci notatnikowej. Operacje, wykorzystujące te układy, a także operacje na danych w rejestrach pamięci i486 są wykonywane dwukrotnie szybciej niż w całym komputerze. Operacje wymagające połączeń z podzespołami na płycie głównej (np. z pamięcią RAM lub pamięcią notatnikową zewnętrzną) są dokonywane z częstotliwością podstawową. Dlatego niezależnie przeprowadzone oceny wykazują przy standardowych zastosowaniach tylko 30–50% przyspieszenie w stosunku do procesora DX.

Komputery zawierające mikroprocesory DX2 stanowią konkurencję dla systemów 80386 SX, 80386 DX, 80486 SX oraz 80486 DX z częstotliwością zegara od 16 do 50 MHz. Wybierając nowy system należy raczej decydować się na najszybszy z aktualnie osiągalnych, gdyż czas oczekiwania na wyniki obliczeń jest najczęściej znacznie cenniejszy niż oszczędności uzyskane w wyniku zakupu przestarzałej technologii. Rozwój oprogramowania narzuca potrzebę coraz większej mocy obliczeniowej. Dlatego wybierając dziś komputer należy uwzględnić oprogramowanie, którym będziemy dysponowali za kilka lat.

Należy zwrócić uwagę, że wyroby konkurencyjne do procesorów firmy Intel mogą mieć korzystniejszy iloraz wydajności i ceny. Wyróżniają się wśród nich układy firmy Advanced Micro Devices (AMD), która według niektórych ocen dostarczyła na rynek 40% sprzedanych procesorów 80386. Na przykład, układ AMD 80386 o częstotliwości zegara 40 MHz jest tańszy od układu Intel i486/25.

PIOTR FUGLEWICZ
Katowice

Języki czwartej generacji a systemy CASE

Języki czwartej generacji (ang. *Fourth Generation* – 4GL) oraz CASE (ang. *Computer Aided Software Engineering*) to dwa terminy pojawiające się ostatnio bardzo często w różnego rodzaju publikacjach, dotyczących projektowania i realizacji systemów przetwarzania danych. Pomimo tego, że historia ich rozwoju jest już, jak na informatykę, dość długa, nadal są one przedstawiane jako nowość i omawiane niezależnie, a czasem nawet we wzajemnej opozycji.

Korzenie technologii związanych z językami czwartej generacji oraz technik stosowanych we współczesnych narzędziach, CASE sięgają przełomu lat sześćdziesiątych i siedemdziesiątych. Podobnie jak było to w przypadku systemu UNIX, który po dekadzie wegetacji w laboratoriach uniwersyteckich stał się w końcu jednym z ważniejszych standardów de facto dla systemów operacyjnych, zarówno CASE jak i 4GL mają za sobą okres życia utajonego.

Tak jak wiele innych pojęć używanych w publikacjach związanych z informatyką, od reklam poczynając, poprzez artykuły w prasie popularnej, po książki wydawane w poważnych wydawnictwach, języki czwartej generacji nie mają jednoznacznej i powszechnie przyjętej definicji. Cechami wspólnymi dla różnych produktów, określanych przez twórców tym mianem są:

- udostępnianie przez język podstawowych własności obecnych w językach poprzednich generacji, takich jak: możliwość budowania iteracji, instrukcja warunku, funkcje i procedury parametryzowane;
- istnienie zdań języka, umożliwiających dostęp na bardzo wysokim poziomie abstrakcji do danych oraz realizacja funkcji kontroli poprawności, spójności i bezpieczeństwa danych,
- występowanie w języku możliwości formatowania i obsługi ekranu,
- możliwość generowania raportów na podstawie danych zawartych w bazach,
- możliwość pracy wielu użytkowników oraz mechanizmy synchronizacji dostępu do danych,
- dostęp do funkcji systemu operacyjnego z poziomu języka,
- istnienie zintegrowanego środowiska rozwoju aplikacji.

Większość znaczących języków klasy 4GL umożliwia również:

- przetwarzanie rozproszone przez obsługę wielu protokołów sieciowych,
- obsługę graficznego trybu komunikacji z użytkownikiem,
- pracę z wieloma bazami zarówno w formacie podstawowym dla danego języka, jak i z bazami innych systemów obsługi baz danych.

Dla lepszego zrozumienia istoty różnicy między 4GL, a językami poprzedniej generacji, przyjrzyjmy się jak różni się obsługa dostępu do danych zapisana w dwóch różnych językach: języku C oraz języku programowania bazy dBase. W przypadku języka C musimy po pierwsze w tym języku lub w języku jeszcze niższego poziomu stworzyć bibliotekę umożliwiającą dostęp do danych zapisanych na nośniku. Z kolei pisząc program korzystamy z procedur tej biblioteki i tworzymy kod, który na przykład może wyglądać tak, jak na wydruku 1.

```
#include "dbf.h"        /* w pliku zdefiniowana jest między
                           innymi struktura DBF, opisująca
                           organizację danych w plikach bazy */

main()
{
 struct DBF *d;          /* zdefiniowanie wskaźnika struktury */
 long record = 1L;       /* określenie numeru rekordu */
 d = (struct DBF *)malloc(sizeof(struct DBF));
                         /* przydzielenie pamięci dla struktury */
 strcpy(d->filename,"dane.dbf"); /* podanie nazwy pliku */
 d_open(d);              /* otwarcie pliku */
 d_getrec(d,record);     /* pobranie rekordu */

 ... modyfikacja zawartości rekordu,
  wykorzystanie danych lub inne operacje ...

 d_putrec(d,record);     /* zapisanie zmienionego rekordu */
 d_close(d);             /* zamknięcie pliku */
 free(d);                /* zwolnienie pamięci */
}
```

Wydruk 1

To samo zapisane w dBase będzie wyglądać: jak na wydruku 2.

```
SELECT A
USE DANE     * otwarcie tablicy danych w jednym z możliwych obszarów
GO TOP       * ustawienie się na pierwszym rekordzie danych

* operacje na danych

CLOSE DATA   * zamknięcie tablicy
```

Wydruk 2

Jak widać, w drugim przypadku kod jest zdecydowanie krótszy i bardziej czytelny, a co za tym idzie, szybszy w pisaniu i łatwiejszy w utrzymaniu. Dodatkowo, kompilator 4GL sprawdza składnię zdań, co częściowo przynajmniej eliminuje niewłaściwe użycie zdań związanych z dostępem do danych, bardziej prawdopodobne przy stosowaniu procedur z osobnej biblioteki.

Programowanie operacji ekranowych i raportów wygląda analogicznie: języki trzeciej generacji wymagają wielu linii kodu dla wyrażenia tego, co w językach czwartej generacji zajmują zaledwie parę linijek. Można więc powiedzieć, że są to narzędzia mające większą siłę ekspresji niż języki poprzednich generacji. W przyrodzie obowiązuje jednak zasada zachowania energii, wyrażana w środowiskach nie naukowych jako „nic za darmo". Za siłę języków czwartej generacji płaci się degradacją wydajności programów stworzonych przy ich użyciu w porównaniu z programami pisanymi w językach niższego poziomu. Wobec rosnącej wydajności sprzętu komputerowego, jeśli uwzględnimy korzyści dotyczące utrzymania i przenośności aplikacji, jest to cena niewielka.

Języki czwartej generacji istnieją zwykle w powiązaniu z własnym systemem zarządzania danymi (*Data Base Management System* – DBMS). DBMS dostarcza usług umożliwiających przechowywanie i dostęp do danych. Coraz powszechniejsza w nowoczesnych 4GL jest możliwość dostępu do danych zapisanych w formacie innego niż własny systemu zarządzania danymi. Na przykład, system PROGRESS umożliwia dostęp do danych zapisanych przy użyciu innych systemów, między innymi RMS i Oracle. Pierwszy z wymienionych systemów obsługuje dostęp indeksowo-sekwencyjny, drugi jest konkurencyjną relacyjną bazą danych. W konsekwencji mamy do czynienia z rozgraniczeniem między 4GL i DBMS, które jest zgodne z modną w ostatnich latach filozofią systemów otwartych.

Ważnym elementem przy tworzeniu systemów otwartych są standardy opisujące zasady przepływu informacji między różnymi środowiskami. Jednym z takich standardów jest SQL (*System Query Language*) opisujący zarówno zasady tworzenia tablic danych, jak i formułowania zapytań. Podobnie jak pozostałe omawiane elementy współczesnych systemów baz danych, SQL został opracowany w początku lat siedemdziesiątych. Stosowanie SQL daje dwie korzyści: pierwszą jest dalsza redukcja kodu programu, drugą – przenośność między różnymi środowiskami.

Skoro posłużyliśmy się przykładem dBase jako języka 4GL, brnijmy dalej i spróbujmy wyrazić w tym języku żądanie wyświetlenia nazwisk wszystkich pracowników, których zarobki przewyższają pewną kwotę. Załóżmy, że mamy dwie tablice: jedną zawierającą dane osobowe, drugą z informacją o dochodach. Kluczem wiążącym te dwie tabele jest identyfikator pracownika.

Program realizujący nasze żądanie może wyglądać jak na wydruku 3.

```
SELECT A
USE PRACOWNI
SELECT B
USE ZAROBKI INDEX IDENT
SELECT A
GO TOP
DO WHILE .NOT. EOF()
  STORE IDENTYFIK TO ROB
  SELECT B
  FIND &ROB
  IF FOUND()
    IF ZAROBKI>200
     ? PRACOWNI->NAZWISKO, ZAROBKI
    ENDIF
  ENDIF
  SELECT A
  SKIP
ENDDO
CLOSE DATA
```

Wydruk 3

Program ten można zapisać nieco zwięźlej, używając na przykład klauzuli *SET RELATION*, wciąż jednak programista musi sam zajmować się organizacją dostępu do tablic danych i ich obsługą. Ta sama operacja zapisana przy użyciu SQL będzie wyglądała jak na wydruku 4.

```
SELECT pracowni.nazwisko, zarobki.zarobki
 FROM pracowni, zarobki
 WHERE (pracowni.identyfik=zarobki.identyfik) AND (zarobki.zarobki>200)
```

Wydruk 4

Jak widać z przytoczonych przykładów, podobnie jak poprzednio, następuje zwiększenie zwięzłości i prostoty języka.

Produkty przeznaczone do tworzenia aplikacji związanych z przechowywaniem i przetwarzaniem dużych zbiorów informacji ewoluują w stronę tzw. ADE (*Application Development Environment*), czyli środowisk składających się z DBMS, 4GL często rozszerzanego o SQL oraz zbioru narzędzi do budowy i utrzymania aplikacji. Jednym z takich narzędzi jest na ogół program umożliwiający pielęgnację struktur bazy danych. W klasycznym relacyjnym modelu danych, a taki stosowany jest najczęściej we współczesnych systemach baz danych, informacja o strukturze danych – tak zwany *metaschemat* – jest również częścią opisu struktury bazy danych. Opis taki utrzymywany jest w słowniku danych (ang. *data dictionary*), często wraz z innymi informacjami. Takimi informacjami mogą być np. treść etykiet, które powinny być domyślnie wyświetlane na ekranie podczas dokonywania operacji na danej lub nazwa funkcji kontrolnej, używanej do kontroli jej poprawności.

Podobny słownik jest na ogół jądrem systemów CASE, które są narzędziem wspomagającym analizę i projektowanie systemów informatycznych. Techniki stosowane w narzędziach CASE są używane w informatyce od ponad dwudziestu lat. Dopiero jednak w latach osiemdziesiątych stworzono środki sprzętowe umożliwiające wykorzystanie tych technik w szerszym zakresie. Narzędzia CASE składają się z dwóch zasadniczych elementów: słownika danych, nazywanego najczęściej repozytorium (ang. *repository*), zawierającego opis aplikacji w wielu różnych aspektach oraz mechanizmów umożliwiających wprowadzanie i redakcję diagramów opisujących te aspekty. Narzędzia CASE dostarczają mechanizmów umożliwiających kontrolę spójności oraz kompletności opisu aplikacji.

We współczesnej praktyce następuje coraz dalej idąca integracja elementów CASE z ADE różnych producentów. Jedni, jak na przykład Oracle, tworzą własne narzędzia, ściśle zintegrowane ze środowiskiem rozwoju aplikacji. Inni, jak PROGRESS, stosując filozofię systemów otwartych, dostarczają pomostów do różnych narzędzi CASE innych producentów. Stosowanie takich pomostów (ang. *bridge*) daje dodatkową korzyść, polegającą na możliwości odtworzenia specyfikacji programu na podstawie analizy istniejącego kodu aplikacji, a także umieszczenie tej specyfikacji w repozytorium, dokonanie poprawek, a następnie generację aplikacji. Takie postępowanie nazywane bywa inżynierią odwrotną (ang. *reverse engineering*).

Pogląd jakoby CASE i 4GL były różnymi klasami narzędzi wydaje się przemijać. Oba elementy uzupełniają się dość dobrze, zwłaszcza biorąc pod uwagę, że obszarem ich działania są różne fazy cyklu życia oprogramowania. O ile CASE koncentruje się na fazach wstępnych, a zarazem najważniejszych dla poprawności przyszłej aplikacji, pokrywając swym zasięgiem analizę problemu i projekt, o tyle 4GL używany jest w fazach implementacji i utrzymania aplikacji.

Równocześnie coraz więcej elementów technik inżynierii oprogramowania jest włączanych w nowe wersje produktów. Można przypuszczać, że ewolucja będzie zmierzać w stronę umieszczania w środowisku rozwoju aplikacji narzędzi umożliwiających graficzne projektowanie. Stanie się to możliwe dzięki coraz powszechniejszemu stosowaniu graficznych form komunikacji z użytkownikiem (ang. *Graphical User Interface* – GUI).

Stosowane obecnie GUI są oparte w większości przypadków na różnym od dotychczasowego podejściu do trybu komunikowania się z użytkownikiem. Podejście obiektowe (ang. *object oriented*) lub inaczej tworzenie aplikacji sterowanych zdarzeniami (ang. *event driven*) stanowi kolejną technikę, która w najbliższych latach znajdzie zastosowanie w programach służących do

dokończenie na s. 25

ANDRZEJ BENDIG-WIELOWIEJSKI
Politechnika Szczecińska
ANDRZEJ RUCIŃSKI
Autokomp Sp. z o.o.
JERZY SARAD
ZETO Szczecin

Zastosowania wielofunkcyjnych elektronicznych kart bezstykowych

Trudno sobie dziś wyobrazić funkcjonowanie obywatela i całych państw bez korzystania z komputerów, sieci teleinformatycznych oraz kart pamięci, spełniających rolę kart identyfikacyjnych, kredytowych itp. Polska wchodząc dziś we wszystkie struktury zorganizowanego świata końca XX wieku ma do przebycia długą drogę dzielącą ją od tego świata w sytuacji cywilizacyjnej, szczególnie widocznej w dziedzinie techniki komputerowej, telekomunikacji i wymiany informacji.

Szansą na dogonienie przez nasz kraj uciekającego świata jest zastosowanie metody przeskoku, a więc wdrażania technologii najnowszych, dopiero pojawiających się w świecie, które zastąpią lub zaczynają zastępować technologie powszechnie dziś stosowane.

W niniejszym artykule autorzy pragną zaprezentować jeden z najnowszych produktów, który jest już wdrażany do praktycznych zastosowań, a wkrótce znajdzie zastosowanie w niemal każdej dziedzinie ludzkiej działalności, w tym również w informatyce. Jest to **wielofunkcyjna elektroniczna karta bezstykowa** zwana po angielsku *Contactless Smart Card*. W dalszej części artykułu będziemy ją nazywać skrótowo kartą CSC.

Idea i cechy elektronicznych kart bezstykowych

Spotykane coraz częściej nazwy: karta elektroniczna, karta pamięci, elektroniczna karta pamięci, elektroniczna karta bezstykowa, inteligentna karta elektroniczna itd. nie są jednoznaczne, bowiem są używane do określenia różnych produktów o różnym przeznaczeniu. Spory bałagan na rynku takich kart ma wiele przyczyn. Przede wszystkim jest to dziedzina na tyle nowa, że nie ugruntowały się jeszcze powszechnie przyjęte potoczne lub komercyjne nazwy. Więcej, nie można jeszcze obecnie mówić o obowiązujących standardach. Jedynie zewnętrzne podobieństwo do popularnych kart magnetycznych (o określonych standardach) skłania do zastąpienia przymiotnika, magnetyczna" nowym określeniem. Warto też zwrócić uwagę na fakt, że pojawienie się kart wykonanych w nowej technologii jest wynikiem poszukiwania doskonalszych rozwiązań w dwóch różnych dziedzinach zastosowań: nośnikach pamięci zewnętrznej komputera oraz osobistych kart identyfikacyjnych. W pierwszym przypadku istotna jest pojemność pamięci, w drugim chodzi o wyeliminowanie fałszerstw i nadużyć.

Karta o dużej pojemności jest traktowana jako następca dyskietki. Np. japońska firma Nippon LSI Card swoje elektroniczne karty bezstykowe o pojemności 8 Mbajtów nazywa *Super FD*. Ze względu na zastosowany nośnik, wyróżniamy karty optyczne (laserowe) i półprzewodnikowe, które z kolei można podzielić, biorąc pod uwagę sprzęg z urządzeniem odczytu zapisu, na karty stykowe i bezstykowe. Niestety ten ogólny podział nie wyczerpuje różnic. M.in. mimo wielu prób normalizacji, nie udało się uzyskać kompatybilności kart wytwarzanych przez różnych producentów.

Karty elektroniczne o wysokim poziomie ochrony dostępu do danych to konkurencja dla kart magnetycznych, które przy swojej niewielkiej pojemności (ok. 100 bajtów) nie mają praktycznie wystarczających mechanizmów zabezpieczających przed kopiowaniem i fałszowaniem. Dynamiczny rozwój kart „inteligentnych" (ang. *Smart Card*) jest nieunikniony [5]. Należy jednak pamiętać o obecnych uwarunkowaniach, które będą miały wpływ na szybkość wdrażania nowej technologii. Wszędzie tam, gdzie istnieje dobrze rozwinięta i sprawdzona infrastruktura wykorzystująca karty magnetyczne o międzynarodowym zasięgu, wprowadzenie kart elektronicznych jako nowego standardu będzie napotykało na opory i wiązało się z dużymi kosztami. Przykładem są tu karty kredytowe, takie jak *Visa, Mastercard, Eurocheque*. Ale mimo to emitenci tych kart myślą już o zmianie, która przede wszystkim wyeliminuje możliwość wykorzystywania kart fałszywych oraz pozwoli na bieżącą kontrolę konta. Szybszych zmian można się spodziewać w systemach bankowych o zasięgu lokalnym (bankomaty). Natomiast najszybszy rozwój nastąpi w dziedzinach, w których dotychczas karty magnetyczne nie były szeroko stosowane, a gdzie karta elektroniczna stwarza nowe możliwości rozwiązania wielu uciążliwych problemów. Następujące cechy kart CSC decydują o szczególnej ich atrakcyjności:

- skuteczna ochrona dostępu do danych na karcie,
- praktycznie brak możliwości podrobienia karty,
- możliwość pracy w ciężkich warunkach (duża wilgotność, wstrząsy itp.),
- długa żywotność (liczba użyć praktycznie ograniczona trwałością zewnętrznej warstwy plastyku),
- odporność na radiację i elektryczność statyczną,
- „inteligencja",
- wielofunkcyjność (jedna i ta sama karta do wielu zastosowań).

Przejście od prostych kart magnetycznych do kart „inteligentnych" można porównać do przełomu, jaki w tej chwili dokonuje się w technologii komputerowej: mikrokomputery ewoluują

w kierunku wieloprocesorowych supermikrokomputerów, których możliwości zdecydowanie zaczynają przewyższać możliwości dużych komputerów (ang. *mainframe*), np. IBM 4381 i niebawem zajmą ich miejsce.

Budowa karty CSC

Zewnętrznie karta CSC przypomina typową kartę plastykową o wymiarach zgodnych z normami ISO (84 × 54 × 1 mm). Wewnątrz zawiera ona zatopioną strukturę półprzewodnikową (zabezpieczoną przed uszkodzeniem mechanicznym metalowym pierścieniem), zintegrowaną z cewkami indukcyjnymi wykonanymi w postaci elastycznego obwodu drukowanego. Cewki służą do indukcyjnego (bezstykowego) przesyłania zasilania i danych do układu elektronicznego, którego struktura jest następująca [4]:

- interfejs sprzęgający z cewkami pozostałą elektronikę,
- układ kodujący,
- nieulotna pamięć danych (EEPROM),
- pamięć programu (ROM),
- pamięć operacyjna (RAM),
- procesor (CPU).

W zastosowaniach, w których nie jest wymagany wysoki poziom zabezpieczenia można stosować tańsze karty bez procesora, wyposażone oczywiście również w pewne mechanizmy ochronne (np. PIN – *Personal Identyfication Number*).

Z obecnie dostępnych kart CSC na uwagę zasługują karty produkowane przez firmy Siemens Nixdorf oraz Philips. W obu przypadkach jest wykorzystywany ten sam bezstykowy interfejs, opracowany przez niemiecką firmę ADE (*Angewandte Digital Elektronik*). Ponadto karty te nie naruszają żadnych dotychczasowych standardów przyjętych dla kart magnetycznych, co pozwala umieścić na karcie CSC również pasek magnetyczny oraz tłoczony numer identyfikacyjny. Daje to możliwość użycia karty także w dotychczasowej infrastrukturze stworzonej dla kart magnetycznych. Fakt ten nie jest bez znaczenia, zwłaszcza w początkowym okresie wprowadzania kart wielofunkcyjnych.

Wielofunkcyjność zastosowań

Możliwości, jakie stwarza inteligentna karta elektroniczna, są często niedoceniane. Próbuje się czasem zastąpić dotychczas stosowaną kartę magnetyczną, żeton lub bilet papierowy zachowując poprzednią logikę funkcjonowania infrastruktury (np. karta telefoniczna z limitowaną liczbą jednostek taryfowych). Takie podejście pozwala co prawda szybko opracować dane zastosowanie, ale jego wdrożenie napotyka na zrozumiały opór rynku. Opór ten wynika z rachunku zysków i strat. Należy pamiętać, że karta CSC jest znacznie droższa od karty magnetycznej. Dlatego akceptacja karty elektronicznej na szerokim rynku jest możliwa dopiero wtedy, gdy jej użycie będzie atrakcyjne zarówno dla użytkownika, jak i dla właściciela infrastruktury. Dlatego w większości projektów opartych na stosowaniu kart elektronicznych zakłada się ich wielofunkcyjność, jedna i ta sama karta jest stosowana w wielu różnych systemach. Oczywiście mogą być odstępstwa od tej reguły w przypadku specjalnych zastosowań, gdzie szczególnie ważny jest poziom bezpieczeństwa, a koszty jego osiągnięcia odnosi się do olbrzymich strat ponoszonych w wyniku braku odpowiedniego systemu zabezpieczającego.

Karta CSC jako elektroniczna portmonetka

Jest to szczególnie ważne zastosowanie [3], które będzie decydować w przyszłości o wielkości produkcji kart elektronicznych. Dobrze rozwinięty na Zachodzie system kart bankowych i kredytowych nie obejmuje swoim zakresem drobnych, często powtarzanych płatności, takich jak komunikacja miejska, taksówki, parkometry, telefony itp. Ponadto uważa się, że operacje finansowe dokonywane za pomocą kart kredytowych stają się nieopłacalne dla banku, jeżeli dotyczą małych kwot. Bariera opłacalności wynosi ponad 50 $ [2]. Obecnie można spotkać tzw. *„prepaid card"*, stosowane głównie w telefonii. Rozwiązuje to tylko problem cząstkowo, zmuszając użytkownika do noszenia przy sobie wielu kart o różnym przeznaczeniu. Karta elektroniczna natomiast może skupić w sobie wszystkie dotychczasowe funkcje przy jednoczesnym znacznym rozszerzeniu zakresu stosowania stając się kartą debetową, przy której pomocy można regulować wszelkie płatności. Niebagatelne znaczenie ma fakt, że karty CSC umożliwiają operacje uzupełniania konta na karcie (czego nie można bezpiecznie czynić na karcie magnetycznej). W ten sposób eliminuje się konieczność wielomilionowych emisji tradycyjnych kart plastykowych, co jest także argumentem ekologicznym, przemawiającym za kartami CSC.

Karta kredytowa

Funkcje karty kredytowej mogą być realizowane według sprawdzonych reguł. Ale i tu wyższość kart CSC jest nie do podważenia, przede wszystkim wyeliminowanie fałszerstw i nadużyć. Drugą bardzo ważną zaletą systemów z kartą CSC jest możliwość autoryzacji karty i sprawdzenie wypłacalności jej właściciela (limitowanie kredytu), bez konieczności korzystania z urządzeń bezpośrednio sprzęgniętych z bankiem. Ta cecha wydaje się szczególnie atrakcyjna przy wdrażaniu systemów w państwach o słabo rozwiniętej sieci telekomunikacyjnej.

Karta CSC jako karta kontrolno-rozliczeniowa

Jest to również bardzo szeroki zakres zastosowań. Należy tu m.in. wymienić:

- jako pierwszy stopień ochrony dostępu do komputera, zbiorów, sieci komputerowych,
- rozliczanie czasu pracy,
- przy kontroli dostępu do pomieszczeń chronionych,
- do rejestracji zdarzeń występujących w pracy urządzeń, np. do rejestrowania parametrów eksploatacyjnych pojazdów samochodowych (czarna skrzynka).

Karta CSC jako karta identyfikacyjna

We wszystkich wyszczególnionych wyżej zastosowaniach funkcje identyfikacyjne są oczywiście wykorzystywane. Tu jednak chodzi o szczególny rodzaj wykorzystania karty elektronicznej jako dokumentu tożsamości, paszportu, prawa jazdy, legitymacji ubezpieczeniowej. Do tych zastosowań predysponuje ją ekstremalnie wysoki stopień zabezpieczenia przed niepowołanym dostępem do danych i sfałszowaniem.

Obecne wdrożenia kart elektronicznych

Od kilku lat obserwuje się olbrzymie zainteresowanie inteligentnymi kartami elektronicznymi. Obecnie prowadzi się na świecie kilkadziesiąt dużych projektów nad nowymi systemami. W większości są to projekty, które opierają się na kartach stykowych, bowiem początkowo nie była jeszcze opanowana technologia kart bezstykowych. Coraz częściej uważa się, że po okresie testów funkcjonalnych tych systemów zostaną one przystosowane do kart bezstykowych. Natomiast nowe przedsięwzięcia zakładają stosowanie wyłącznie kart bezstykowych.

Do ciekawszych realizacji należy zaliczyć [3, 4]:

- włoski projekt pieniądza elektronicznego o nazwie *Moneta*,
- płatności w transporcie miejskim w Manchesterze, w Danii i Finlandii,
- płatności za korzystanie z energii w Wielkiej Brytanii,
- elektroniczna portmonetka, opracowana we Francji.

Należy tu wspomnieć o polskim projekcie (AK-SYSTEM) wielofunkcyjnej elektronicznej karty bezstykowej (MAXI-CARD), który od końca 1990 r. jest rozwijany przez firmę Autokomp. W ramach AK-Systemu jako pierwszy powstał projekt dla komunikacji miejskiej [2] (pilotażowa instalacja testowana od końca 1991 r.). Obecnie są rozwijane kolejne zastosowania tworzące spójną infrastrukturę dla kart wielofunkcyjnych. System rejestracji czasu pracy i ochrony dostępu do komputerów oraz sieci komputerowych jest realizowany wspólnie ze szczecińskim ZETO (tam też jest wdrażany). Obie firmy dzięki zaawansowaniu swoich badań zostały włączone w prace nad podobnymi systemami prowadzonymi na Zachodzie (C2-WIN, MFK-Projekt).

Poniżej przedstawiono funkcje jednego z rozwijanych systemów – systemu ochrony dostępu do komputera i informacji.

System ochrony dostępu do komputera i informacji

Jest to system kontroli dostępu do komputera, sieci komputerowych oraz programów i baz danych, wykorzystujący karty CSC. Zamontowany w komputerze czytnik kart (tzw. kieszeń CSC) uniemożliwia dostęp do komputera bez użycia karty.

System ten zapewnia:
- niezawodne, całkowite zabezpieczenie przed dostępem do danych i programów przez osoby nieupoważnione,
- zabezpieczenie przed przypadkowym lub celowym zniszczeniem zbiorów przez osoby nieupoważnione,
- ułatwienie obsługi systemów komputerowych,
- możliwość ustalenia ostatniego użytkownika oraz czasu jego pracy, nawet po zniszczeniu wszystkich zbiorów pamięci (czarna skrzynka).

Moduł dla sieci NOVELL NETWARE

Funkcje systemu:
- kontrola włączania i wyłączania serwera zbiorów (ang. *file server*),
- blokowanie i odblokowywanie stacji dysków elastycznych serwera sieci,
- ograniczenie lub całkowite blokowanie komend wydawanych z konsoli serwera sieci,
- kontrola włączania i wyłączania stacji roboczych,
- pełna ochrona lokalnych zbiorów w stacjach roboczych.

Rodzaje stosowanych kart:
KWS – Karta Właściciela Sieci, ma wszystkie uprawnienia. Używana w wyjątkowych sytuacjach (normalnie jest zamknięta w sejfie), takich jak instalowanie sieci, zmiana hasła sieci, inicjowanie karty administratora sieci;
KAS – Karta Administratora Sieci, ma wszystkie uprawnienia z wyjątkiem: instalacji sieci, zmiany hasła sieci;
KOS – Karta Operatora Sieci, jej uprawnienie nadaje nadzorca sieci. W szczególnym przypadku uprawnienia mogą być ograniczone do włączenia i wyłączenia serwerów sieci;
KUS – Karta Użytkownika Sieci, uprawnia do korzystania z wybranych stacji roboczych. W szczególnym przypadku może łączyć uprawnienia karty użytkownika komputera autonomicznego.

Moduł dla komputera autonomicznego

Funkcje systemu:
- kontrola włączenia i wyłączenia,
- blokowanie i odblokowywanie stacji dysków twardych i elastycznych,
- blokowanie poleceń systemowych (np. kopiowanie, kasowanie zbiorów),
- nadawanie uprawnień dostępu do zbiorów,
- rejestracja użytkowników i ich czasu pracy.

Rodzaje kart:
KWK – Karta Właściciela Komputera, ma wszystkie uprawnienia. Używana w wyjątkowych sytuacjach (normalnie zamknięta w sejfie), takich jak: konfigurowanie komputera (*setup*), formatowanie dysków, inicjowanie karty administratora komputera;
KAK – Karta Administratora Komputera, ma wszystkie uprawnienia z wyjątkiem konfigurowania komputera i formatowania dysków;
KUK – Karta Użytkownika Komputera, jej uprawnienia określa nadzorca komputera. W szczególnym przypadku karta może upoważniać do korzystania tylko z jednego programu użytkowego ładowanego automatycznie.

System ochrony dostępu do komputera można zintegrować z innymi systemami wykorzystującymi karty CSC, np. z systemem kontroli dostępu do pomieszczeń chronionych, systemem rejestracji czasu pracy (RCP) i innymi, opartymi na kartach CSC.

LITERATURA

[1] Bendig-Wielowiejski A.: Contactless Smart Cards in Information Systems. International Scientific School ISAT'92, Karpacz, September 1992
[2] Chocianowicz W., Ruciński A.: AK-SYSTEM w komunikacji miejskiej. Biuletyn informacyjny IGKM nr 4, Warszawa, kwiecień 1992
[3] Chocianowicz W., Ruciński A.: With Contactless Chip Card from Public Transportation to the Universal Purse. ESKAT'92, Finlandia, September 1992
[4] Materiały konferencyjne Smart Card '92. London 3–5 March 1992
[5] Watts A.: Mikrokomputer in Smart-Card-Anwendungen. Mikroelektronik. Bd. 6, 1992, Heft 2.

Przypominamy, że Zakład Wydawniczy „Elektronizacja" wydaje w I kwartale 1993 r. bardzo poszukiwany, najobszerniejszy i najnowocześniejszy na polskim rynku

SŁOWNIK SKRÓTÓW ANGIELSKICH STOSOWANYCH W ELEKTRONICE, INFORMATYCE I TELEKOMUNIKACJI

W związku z nadspodziewanie dużym zainteresowaniem tym wydawnictwem postanowiono zwiększyć jego nakład. Słownik można otrzymać pocztą pod wskazany adres, wpłacając preferencyjną, zniżkową kwotę 50.000 zł w terminie trzech tygodni od daty ukazania się niniejszej informacji. **Konto: PBK III O/Warszawa nr 370015-1573-139-11. Wydawnictwo SIGMA-NOT sp. z o.o., Zakład Kolportażu, 00-950 Warszawa, skr. poczt. 1004.** Zamawiając słownik w przedpłacie, masz gwarancję otrzymania go po cenie ulgowej, bez wysokiej marży sieci handlowej i kosztów wysyłki.

JAROSŁAW A. BERENT
Akademia Medyczna
Bydgoszcz

TIFF – standard grafiki pikselowej

Celem niniejszego artykułu jest przybliżenie Czytelnikowi standardu TIFF (ang. *Tag Image File Format* – Etykietowy Format Plików Obrazu), który najprawdopodobniej w najbliższym czasie stanie się wiodącym sposobem przechowywania, obsługi i modyfikacji grafiki komputerowej. Prawa autorskie do niego są własnością firmy Aldus Corporation, Seattle, USA.

Format TIFF może być zastosowany wszędzie tam, gdzie użytkownik wymaga od programu obsługi grafiki pikselowej. Został on zdefiniowany jako format o bardzo szerokim stosowaniu i z łatwością może być zastosowany w prostych programach lub urządzeniach zewnętrznych. Istnieje również opcja rozszerzania struktury pliku dla realizowania specjalnych rozwiązań indywidualnego użytkownika.

Budowa plików TIFF jest ściśle zdefiniowana i niezależna od typu komputera, a wynika tylko z aktualnych potrzeb użytkownika.

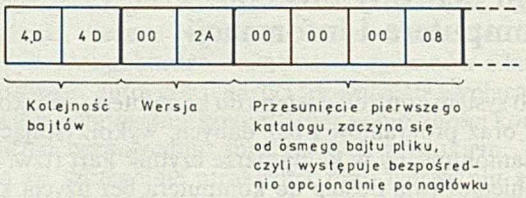
Według zaleceń autorów formatu użytkownik powinien stosować rozszerzenia nazwy *.TIF. Maksymalna, teoretycznie dozwolona długość pojedynczego pliku wynosi 4 GB, który jest traktowany jako sekwencja ośmiobitowych bajtów.

Budowa pliku

Plik rozpoczyna się nagłówkiem i zawiera jeden lub więcej katalogów złożonych z pól. Przed, po lub między katalogami znajduje się informacja składająca się na obraz. Jeden plik może zawierać jeden obraz wyjściowy oraz opcjonalnie jego warianty. Pojedynczy obraz może być zapisany w pliku w jednym ciągu lub w wielu fragmentach, rozmieszczonych w różnych miejscach pliku.

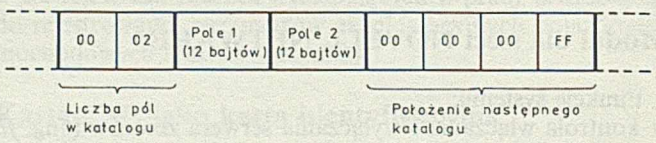
Obraz jest prostokątną tablicą piksli. Informacja o pojedynczym pikslu jest sekwencją bitów o długości określonej przez iloczyn wartości pól: *Bits_per_sample* × *Samples_per_pixel*. Wymiary obrazu w pikslach wynikają z wartości pól: *Image_lenght* (wysokość) i *Image_width* (szerokość). Obraz może być podzielony na poziome pasy (ang. *strip*) w celu łatwiejszego dostępu do wybranego fragmentu obrazu. O liczbie pasów mówi wartość pola: *Strip_offsets*, a podane tam przesunięcia wskazują na ich położenie w pliku. Każdy pas składa się z linii (ang. *Row*). Liczba linii tworzących pas jest podana w polu: *Rows_per_strip*.

Każdy plik TIFF rozpoczyna się ośmiobajtowym nagłówkiem (ang. *Image File Header*). Pierwsze dwa bajty określają sposób zapisu wewnątrz pliku. Dla zapisu od najmniej do najbardziej znaczącego bajtu zawierają one wartość 4949*hex*, dla kolejności obrotowej 4*D*4*Dhex*. Dwa kolejne bajty, pierwotnie pomyślane jako miejsce na oznaczenie wersji TIFF-u, faktycznie zawierają zawsze wartość 2*Ahex*. Może to być wykorzystane do programowego rozpoznawania plików tego typu. Kolejne cztery bajty określają położenie pierwszego katalogu (ang. *Image File Directory*) przez podanie tzw. przesunięcia (ang. *offset*). Przesunięcie jest to adres odnoszący się do początku pliku. Przykładowo, jeżeli dany plik rozpoczyna się sekwencją: 4*D*4*D*_002*A*_00000008, oznacza to, że pierwszy katalog wystąpi bezpośrednio po nagłówku. Oczywiście tę samą informację można przekazać jako: 4949_2*A*00_08000000 (rys. 1).



Rys. 1. Budowa nagłówka

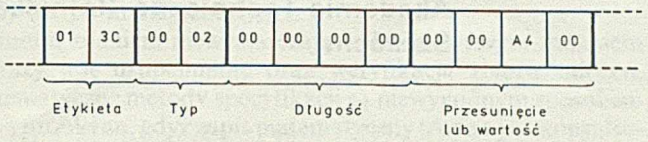
Każdy katalog jest sekwencją bajtów o długości 2 + n * 12 + + 4. Pierwsze dwa bajty zawierają wartość *n*, która oznacza liczbę dwunastobajtowych pól (ang. *field entry*), a ostatnie cztery – przesunięcie kolejnego katalogu. Jeżeli dany katalog jest ostatnim, to cztery końcowe bajty zawierają 00000000*hex*. Następujące po sobie dwunastobajtowe sekwencje, począwszy od trzeciego bajtu, są to pola (rys. 2).



Rys. 2. Budowa katalogu
Powyższy katalog ma długość 2 + 2 × 12 + 4 = 30 bajtów oraz dwa pola. Ostatnie cztery bajty informują, że następny katalog rozpoczyna się od adresu FF.

Pole zawiera dane na temat jednej cechy obrazu zawartego w pliku. Całkowita jego długość wynosi 12 bajtów. Dwa pierwsze zawierają numer etykiety (ang. *tag*) od 0000*hex* do *FFFFhex*. Praktycznie jest wykorzystane jedynie 45, ale teoretyczna możliwość 65 535 wartości znakomicie oddaje ekspansywność standardu. Pola w katalogu powinny być posortowane według rosnących wartości etykiet. Dwa kolejne określają typ pola. Wartości 1 odpowiada typ *BYTE* (8-bitowa liczba całkowita bez znaku), 2 to typ *ASCII* (8-bitowy, zawiera znaki ASCII i dodatkowo ostatni bajt zawierający NUL), 3 to typ *SHORT* (dwubajtowa, szesnastobitowa liczba całkowita bez znaku), 4 to typ *LONG* (jak SHORT, ale 4-bajtowa) i 5 to typ *RATIONAL*, składający się z dwóch *LONG*. Cztery kolejne bajty stanowią długość pola w jednostkach wynikających z typu. Przykładowo, pojedyncze słowo typu RATIONAL ma długość 1, a nie 4. Ostatnie cztery bajty pola zawierają albo wartość, jeżeli mieści się ona w czterech bajtach, albo przesunięcie wartości, pozwa-

lające na jej odnalezienie w pliku. O tym co faktycznie zawierają te bajty, decyduje typ i długość pola. Ten może zbyt skomplikowany opis wyjaśnia rys. 3.



Rys. 3. Budowa pola
Powyższe pole ma etykietę 13Chex, typu ASCII, długość tekstu umieszczonego od adresu A400hex wynosi Dhex = 13 dec (12 znaków + NUL na końcu).

W pliku może być więcej niż jeden katalog. Jedną z możliwości wykorzystania tego faktu jest stworzenie w ramach jednego pliku kilku wersji obrazu wyjściowego. Na przykład, pierwszy katalog może wówczas opisywać sposób organizacji obrazu wyjściowego, a katalogi kolejne – definiować wersję czarno-białą, wersję o zmienionym kontraście, wymiarach itd.

Pola

Kompletny opis standardu zawiera definicje kilkudziesięciu pól. Część z nich jest konieczna do definiowania obrazu, inne są opcjonalne. Kompletny opis wszystkich pól przekracza ramy niniejszego artykułu. Przedstawię jedynie kilka, co pozwoli skonstruować w następnym rozdziale przykładowy plik. Jest to moim zdaniem niezbędne dla całkowitego zrozumienia tekstu.

Pole *BITS_PER_SAMPLE*
Etykieta: 102*hex*
Typ: *SHORT*
Wartość: *Samples_per_pixels*
Definicja: ile bitów powinno opisywać każdą wartość piksla obrazu.

Pole *IMAGE_LENGHT*
Etykieta: 101*hex*
Typ: *SHORT* lub *LONG*
Wartość: 1
Definicja: wysokość obrazu w pikslach.

Pole *IMAGE_WIDTH*
Etykieta: 100*hex*
Typ: *SHORT* lub *LONG*
Wartość: 1
Definicja: szerokość obrazu w pikslach.

Pola *PHOTOMETRIC_INTERPRETATION*
Etykieta: 106*hex*
Typ: *SHORT*
Wartość: 1
Definicja: 0 dla obrazów czarnobiałych, 0 jest traktowane jako biel, a 255 jako czerń.

Pole *ROWS_PER_STRIP*
Etykieta: 116*hex*
Typ: *SHORT* lub *LONG*
Wartość: 1
Definicja: liczba linii tworzących jeden pas.

Pole *SAMPLES_PER_PIXEL*
Etykieta: 115*hex*
Typ: *SHORT*
Wartość: 1
Definicja: liczba wartości określających pojedynczy piksel.

Pole *STRIP_OFFSETS*
Etykieta: 111*hex*
Typ: *SHORT* lub *LONG*
Wartość: liczba pasów obrazu
Definicja: podaje przesunięcia kolejnych pasów tworzących obraz.

Pole *HOST_COMPUTER*
Etykieta: 13*Chex*
Typ: *ASCII*
Definicja: nazwa komputera programisty.

Przykładowy plik TIFF

Adres	Bajty pliku	Uwagi
00	4*D*4*D*	Kolejność bajtów na pliku
02	002*A*	Wersja
04	00000008	Pierwszy katalog
08	0008	Jest w nim 8 pól
0*A*	0100	Pole 1: *Image_width*
OC	0003	Typ: *SHORT*
0*E*	00000001	Wartość: 1
11	00000006	Obraz szeroki na sześć piksli
14	0101	Pole 2: *Image_length*
16	0003	Typ: *SHORT*
18	00000001	Wartość: 1
1*C*	00000006	Obraz wysoki na sześć piksli
20	0102	Pole 3: *Bits_per_sample*
22	0003	Typ: *SHORT*
24	00000001	Wartość: 1
28	00000008	8 bitów opisuje każdą wartość piksla
2*C*	0106	Pole 4: *Photometric_interpretation*
2*E*	0003	Typ: *SHORT*
30	00000001	Wartość: 1
34	00000000	Czarno-biały, 0-biel, 255-czerń
38	0111	Pole 5: *Strip_offsets*
3*A*	0004	Typ: *LONG*
3*C*	000000003	Obraz ma trzy pasy
40	000000079	Położenie przesunięć pasów
44	0115	Pole 6: *Samples_per_pixel*
46	0003	Typ: *SHORT*
48	000000001	Wartość: 1
4*C*	000000001	Każdy piksel opisuje jedną wartość
50	0116	Pole 7: *Rows_per_strip*
52	0003	Typ: *SHORT*
54	000000001	Wartość: 1
58	000000002	Pas składa się z dwóch linii
5*C*	013*C*	Pole 8: *Host_computer*
5*E*	0002	Typ: *ASCII*
60	00000000*D*	Tekst ma 12 + 1 znaków
64	00000006*C*	Położenia tekstu na pliku
68	000000000	Nie ma następnego katalogu
6*C*	494277205067	IBM PC
72	203438364458	486*DX*
78	00	*NUL* kończący wyrażenie
79	00000085	Położenia pierwszego pasa
7*D*	0000009*C*	Położenia drugiego pasa
81	00000090	Położenie trzeciego pasa
85	000000000000	Pierwszy pas
8*B*	00*FFFFFFFF*00	Pierwszy pas cd.
90	00*FFFFFFFF*00	Trzeci pas
96	000000000000	Trzeci pas cd.
9*C*	00*FFFFFFFF*00	Drugi pas
*A*2	00*FFFFFFFF*00	Drugi pas cd.
*A*8		EOF

Powyższy przykładowy plik TIFF składa się z *A*8*hex* bajtów. Tworzy on czarno-biały obraz o wymiarach sześć na sześć piksli, ma trzy pasy po dwie linie. Każdy piksel jest opisany przez jedną wartość 8-bitową. Pasy są położone w kolejności: pierwszy, trzeci i drugi. Przedstawia on czarny kwadrat cztery na cztery piksle, położony w środku pola.

dokończenie na s. 25

JAROSŁAW BOCHIŃSKI
Zakład Zastosowań Matematyki
Instytut Matematyki i Fizyki
Akademia Techniczno-Rolnicza
Szczecin

Matematyka w tworzeniu oprogramowania?

Zastanawiam się często, czy jesteśmy przygotowani do „inwazji" matematyki i matematyków w tak praktyczną działalność, jaką jest tworzenie oprogramowania?

Trudno jest nie doceniać wpływu matematyki na rozwój i stan obecny metod i narzędzi tworzenia oprogramowania. Z drugiej jednak strony dostrzegam uśmiechy praktyków, gdy mowa o praktycznym wykorzystaniu matematycznych metod projektowania i tworzenia aplikacji.

Moim zdaniem powody takich reakcji są następujące:
- matematyka operuje własnym, ścisłym językiem, często niezrozumiałym dla niematematyków,
- formalne specyfikowanie problemu wydaje się być bardzo czasochłonne i nipotrzebne wobec istnienia wielu silnych narzędzi projektowania i programowania (metody strukturalne, języki czwartej generacji, generatory aplikacji, programy uruchomieniowe itp.).

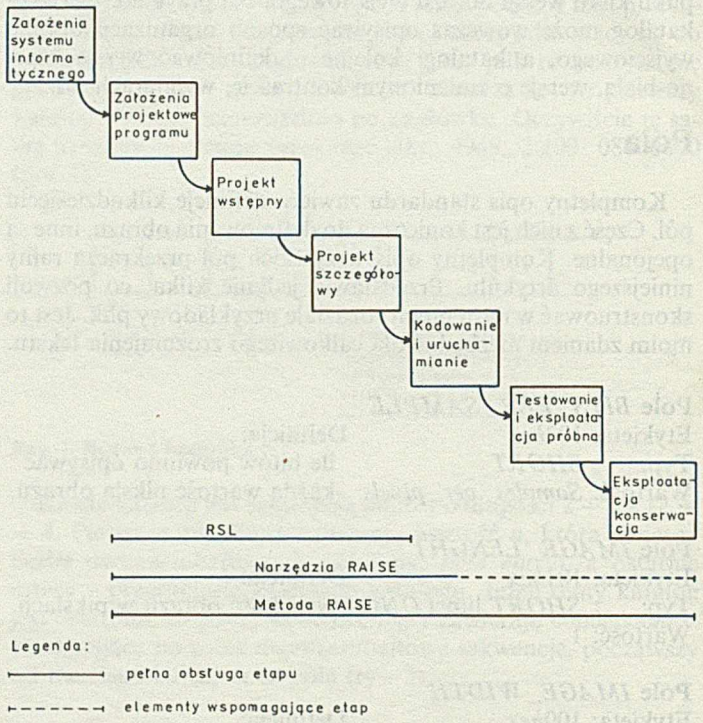
Sądzę, że zaprezentowanie konkretnej metody będzie ciekawsze niż dyskutowanie wymienionych problemów. Opiszę w tym celu RAISE – jedną z metod komercyjnych (w odróżnieniu od wielu często bardzo efektownych prac o charakterze teoretycznym), którą zaprezentowano na międzynarodowych konferencjach VDM'91 oraz FORTE'92 [3].

RAISE – metoda, język i narzędzia

Metoda RAISE (*Rigorous Approach to Industrial Software Engineering*) zakłada obfite wykorzystanie metod matematycznych w procesie tworzenia oprogramowania. Celem ostatecznym jest system informatyczny niezawodny, poprawny, dobrze udokumentowany i łatwy w konserwacji dzięki wykorzystaniu formalnej (matematycznej) notacji – języka RSL (*RAISE Specification Language*) [2] oraz metodologii i narzędzi wspomagających kolejne fazy życia programu (rys.).

Charakterystyka metody

Metoda RAISE wywodzi się z VDM (*Vienna Development Method*) – najpopularniejszej ze znanych metod i notacji formalnych, jednakże w stosunku do VDM jest wzbogacona o:
- zbiór narzędzi programowych,



System informatyczny wykorzystujący matematyczną notację języka RSL oraz metodologię i narzędzia wspomagające kolejne fazy życia programu

- wspomaganie wielu kroków rozwoju programu,
- elementy istotne przy tworzeniu dużych systemów, takie jak zapewnienie odpowiedniej jakości oraz konserwacji systemu.

Metoda RAISE polega na stopniowym rozwijaniu projektu w następujący sposób:
- program jest konstruowany w serii kroków,
- każdy krok rozpoczyna się od opisu programu i tworzy nowy opis, który jest bardziej szczegółowy, a jednocześnie zgodny z opisem poprzednim (może więc go zastąpić),
- rozwijane są jednocześnie algorytmy i struktury danych,
- początkowo abstrakcyjny (deklaratywny) opis programu staje się coraz bardziej konstruktywny i zbliżony do implementacji,
- poprawność kolejnych kroków może być formalnie dowodzona,
- w ostatnim kroku jest generowany program lub zespół programów w wybranym języku programowania za pomocą translatorów podzbioru RSL do języków programowania (Ada, Modula 2, C++).

Formułowanie założeń dla programu

Początkowy opis wymagań stawianych systemowi informatycznemu jest formułowany zazwyczaj w języku naturalnym.

Mgr inż. JAROSŁAW BOCHIŃSKI ukończył w 1988 r. studia na Uniwersytecie Gdańskim – Matematykę ze specjalnością Metody Numeryczne i Programowanie. Pracuje w Zakładzie Zastosowań Matematyki w Instytucie Matematyki ATR w Bydgoszczy oraz w przedsiębiorstwie informatycznym „Saba". Interesuje się formalnymi metodami projektowanie programów.

Zalety wykorzystania na tym etapie języka formalnego są niewątpliwe, ponieważ:
- prezentowanie informacji odbywa się w jednorodnej formie,
- język taki stanowi punkt początkowy dla kolejnych etapów pracy (aż po implementację),

ale moim zdaniem niewystarczające. Podstawowym zadaniem tej fazy jest uzupełnianie oraz weryfikacja zbioru założeń. Matematyczne metody specyfikacji są niewygodnym sposobem opisu problemu, gdyż zapis matematyczny trudno jest konsultować z użytkownikiem programu, a nawet z jego konsultantem. Dlatego wskazane byłoby połączenie w jedną całość znanych już od dłuższego czasu metod „strukturalnych" oraz metod formalnych. Jak dotąd, mimo wyraźnie rysującej się potrzeby, nie spotkałem takiego zestawienia metod i narzędzi.

Faza projektowa

W metodzie RAISE głównym etapem tworzenia systemu jest projektowanie. Polega ono na ciągłym rozwijaniu specyfikacji, od fazy początkowej, przez serię kolejnych kroków, aż do fazy implementacji systemu. W końcu tej fazy wykrystalizowuje się ostateczna postać programu.

Każdy krok projektowy może polegać na:
- konkretyzowaniu specyfikacji przez:
  - zastępowanie abstrakcyjnych typów danych typami konkretnymi,
  - wymianie algorytmu na bardziej efektywny (np. przy zastosowaniu procesów współbieżnych);
- rozszerzaniu specyfikacji o nowe elementy (funkcje lub procesy);
- dzieleniu specyfikacji na części w celu ich samodzielnego rozwijania;
- łączeniu przygotowywanych osobno specyfikacji;
- podstawianiu aktualnego parametru do wcześniej sparametryzowanej specyfikacji.

Implementacja

Przygotowaną w fazie projektowej specyfikację można przetłumaczyć automatycznie na wybrany język programowania. RAISE zapewnia translatory tłumaczące podzbiór elementów RSL na konstrukcje języków C++, Ada i Modula 2. Dyskusyjna może być optymalność automatycznej implementacji, choć mamy na nią duży wpływ przez dobór w fazie projektowej odpowiednich algorytmów i struktur danych.

W metodzie RAISE proces tworzenia programu, to głównie proces projektowy, a nie – jak w dotychczasowym podejściu do programowania – żmudne kodowanie projektu. Oczywiście wymaga to innych kwalifikacji zespołu projektowego.

Nie potrafię obecnie odpowiedzieć na takie pytania, związane z implementacją projektu zapisanego w języku RSL, jak:
- czy translatory RSL pozwalają na wykorzystanie bibliotek narzędziowych, a jeśli tak, to jakich?
- czy można wykorzystać standardowe mechanizmy zarządzania DBMS w projektach realizowanych w RAISE, a jeśli tak, to (SQL?) oraz w jaki sposób?

Spodziewam się, że autorzy (na których odpowiedź obecnie czekam) przewidzieli odpowiednie, satysfakcjonujące mechanizmy.

Rejestracja kolejnych kroków rozwoju specyfikacji

Dzięki mechanizmom rejestracji kolejnych stadiów rozwoju, wersji i wariantów specyfikacji metoda RAISE wspomaga:
- zarządzanie projektami (większość pracy polega na rozwijaniu specyfikacji),
- prace konserwacyjne (nie tylko programu, ale również całości specyfikacji),
- ponowne użycie części specyfikacji w nowych projektach.

Jakość programu

Matematyczne podstawy języka RSL i metody RAISE oraz zbiór narzędzi pozwalają na:
- konstruowanie specyfikacji, które są dobrze przemyślane, abstrakcyjne, łatwe do czytania i odpowiednio udokumentowane,
- kontrolę:
  - składni i stylu oraz statyczną kontrolę typów (rekursywnych),
  - błędów w rodzaju: dzielenie przez zero, zakleszczanie (ang. *deadlock*), błędy rekursji,
- sprawdzanie czy są spełniane założenia w całym procesie rozwijania specyfikacji,
- weryfikację kolejnych kroków przekształcania specyfikacji.

Konserwacja programu

Jedną z korzyści stosowania metody RAISE jest łatwość konserwacji i rozwijania produktów programowych. Pomoc w tych pracach polega na:
- mechanizmach kontroli wersji i wariantów specyfikacji,
- automatycznego przenoszenia poprawek i zmian do kolejnych wersji,
- zarządzaniu i dokumentowaniu całości prac projektowych.

Język specyfikacji RAISE

Język RSL jest bardzo bogatym językiem, którego konstrukcje pozwalają na jego wykorzystanie na wszystkich etapach rozwoju projektu, zarówno we wstępnej fazie abstrakcyjnej i ogólnej specyfikacji, jak i na etapie zorientowanego implementacyjnie opisu.

Główne cechy RSL

Język RSL realizuje główne kierunki rozwoju formalnych specyfikacji z okresu ostatnich dwudziestu lat, takich, jak:
- specyfikacja algebraiczna,
- specyfikacja zorientowana modelowo,
- modularyzacja i parametryzacja,
- definicje, zarówno aksjomatyczne, jak i explicite,
- style: aplikatywny, imperatywny i równoległy.

Dzięki takiej różnorodności łatwo jest dobrać styl odpowiedni do potrzeb i upodobań na każdym poziomie projektu.

Moduły

Moduły są podstawowymi elementami specyfikacji metody RAISE. Można je porównać do takich konstrukcji znanych języków programowania, jak klasy, obiekty, moduły lub pakiety. Wewnątrz modułów są definiowane następujące elementy języka RSL:
- typy (w znaczeniu zbiorów),
- wartości (odpowiedniki funkcji, konstruktorów i destruktorów obiektu),
- zmienne (odpowiedniki pól obiektu),
- kanały (dla komunikowania procesów równoległych).

Moduł podczas definiowania własnych elementów może odwoływać się do elementów innych modułów (np. przez dziedziczenie). Parametryzowanie modułów powoduje, że w za-

leżności od dostarczonych parametrów mogą być one zastosowane w różnych kontekstach.

Typy, wartości i zmienne

RSL dostarcza odpowiedni zbiór typów atomowych (*Bool, Int, Text, Char* itp.) oraz konstruktorów typów (funkcji, ciągów, zbiorów, rekordów itp.). Każdy z typów złożonych jest traktowany na równi z typem atomowym (np. istnieją zmienne tego typu, może on być dziedziną lub przeciwdziedziną funkcji), odmiennie niż w większości języków programowania. RSL pozwala na definiowanie typów abstrakcyjnych, dla których nie precyzuje się zbioru elementów.

Wartości (funkcje) modułu można definiować explicite lub za pomocą aksjomatów opisujących ich własności. Zmienna zadeklarowana w module służy do przechowywania informacji o stanie modułu (analogicznie do stanu automatu, lub pola obiektu).

Procesy

Działanie równoległe może być wyspecyfikowane w języku RSL za pomocą procesów. Proces to element modułu, który może komunikować się z innymi procesami przez kanały oraz zmienne. Pojęcie procesów równoległych w RSL opiera się na algebrze procesów CCS i CSP. Mogą być one również opisane explicite lub aksjomatycznie.

Narzędzia RAISE

Narzędzia RAISE stanowią spójny zbiór, pokrywający większość czynności w cyklu życia programu. Są one skoncentrowane wokół biblioteki RAISE, która jest specjalistycznym systemem baz danych.

Środowisko RAISE jest przygotowane do zastosowania w dużych organizacjach projektowych, wytwarzających złożone systemy oraz zatrudniających wielu projektantów i programistów.

Współczesne wymagania sprzętowe i programowe to:
- stacje robocze Sun lub IBM 6000,
- system operacyjny Unix,
- system X Windows lub SunView,

Planowane jest przygotowanie wersji dla systemu PS/2 AD/ /Cycle.

Biblioteka

Biblioteka RAISE jest zbudowana z wykorzystaniem systemu Oracle rdbms i służy do przechowywania następujących elementów:

- modułów – z kontrolą wersji oraz odwołań do innych modułów,
- relacji między modułami,
- teorii – zbiorów predykatów zapisanych w RSL, prawdziwych dla jednego lub wielu modułów,
- kroków rozwoju specyfikacji,
- dowodów – dotyczących w pełni formalnych lub nieformalnych kroków rozwojowych,
- konfiguracji specyfikacji – opisu aktualnie wykorzystywanego do stworzenia aplikacji zbioru modułów,
- wskazówek.

Zmiany w poszczególnych wersjach mogą być przenoszone do kolejnych wersji na życzenie lub w pełni automatycznie.

Edytor modułów i inne edytory

Edytor modułów jest przystosowany do tworzenia nowych modułów lub ich kolejnych wersji. W części jest on sterowany składnią RSL, a w części funkcjonuje jak zwykły edytor tekstów. Zapewnia syntaktyczną poprawność modułów oraz przeprowadza statyczną kontrolę typów.

Praca z tego typu edytorem wymusza standardowy wygląd specyfikacji i znacznie przyspiesza prace edycyjne. Podobne własności mają inne edytory RAISE: relacji, kroków rozwoju, konfiguracji, wskazówek, dowodów oraz teorii.

Narzędzia weryfikacji

Zestaw narzędzi weryfikacji pozwala użytkownikowi na generowanie tez, które wymagają weryfikacji, oraz wspomaga sam proces dowodzenia. W jego skład wchodzą:
- edytor dowodów – przeznaczony do wspomagania formalnych dowodów poprawności kolejnych kroków rozwojowych,
- animator – pozwalający na symboliczne uruchomienie podzbioru języka RSL.

Oczywiście nie należy liczyć na w pełni automatyczne narzędzia dowodzenia; dowody są tworzone w interakcji z projektantem, wybierającym najwłaściwszą drogę spośród możliwych i zaproponowanych przez edytor dowodów.

Translatory

Obecnie są oferowane translatory dla języków programowania Ada oraz C++.

Zazwyczaj ręczna transkrypcja specyfikacji niskiego poziomu do kodu jest całkiem prosta, jednakże często niebezpieczna z uwagi na potencjalną możliwość powstania błędów. Ponadto konserwacja programu przetranslowanego ze specyfikacji (poprawki i zmiany) jest bardziej efektywna i dobrze udokumentowana.

Stan obecny oraz przyszłość RAISE

W ramach projektu LaCoS ESPRIT-II, który jest realizowany od 1990 r., następujące organizacje wykorzystują RAISE do tworzenia złożonych systemów informatycznych: Bull SA, MATRA Transport SA, Technisystems Ltd, Lloyd's Register of Shipping, Space Software Italia SA oraz Inisel Espacio SA.

Jednocześnie duńska firma CRI A/S (*Computer Resources International*), przygotowała komercyjną wersję narzędzi RAISE, która obecnie jest rozwijana o nowe elementy [1, 4].

* * *

RAISE jest metodą formalnej specyfikacji, jak dotąd najlepiej wyposażoną w narzędzia, o najszerszym podłożu, wspomagającą wiele różnych stylów specyfikacji i stosowaną w praktyce już od dwóch lat. Już pierwszy kontakt pozwala docenić jej walory dzięki bardzo przyjaznemu środowisku pracy projektanta.

W metodzie tej jest odczuwalny jednak brak interfejsu do strukturalnych metod projektowania systemów informatycznych, które dzięki możliwości graficznej prezentacji są czytelne dla szerokiego spektrum użytkowników. Takiego związku z metodami strukturalnego projektowania nie ma również żadna inna ze znanych mi metod formalnych. Poza tym istotnym brakiem, istnieją również czysto praktyczne trudności

w wykorzystaniu w praktyce projektowej metod formalnych, a mianowicie:

- brak wykwalifikowanej grupy zawodowej matematyków – projektantów,
- niski poziom organizacyjny przedsiębiorstw informatycznych,
- brak przekonania do rzeczywistych korzyści płynących z zastosowania metod formalnych, zarówno wśród kadry kierowniczej jak i programistów.

Pomimo tych niedostatków metod formalnych sądzę, że na postawione w tytule artykułu pytanie, można odpowiedzieć twierdząco.

LITERATURA

[1] Eriksen K. E., Prehn S.: RAISE Overview. CRI A/S, październik 1991
[2] George C.: RAISE Specification Language. Prentice-Hall International, 1991
[3] George C.: The RAISE Specification Language: A Tutorial. Proceedings 4th International Symposium of VDM Europe, Vol. 2, p. 238–320, 21–25 październik 1991
[4] RAISE Tools Snapshots. CRI A/S, październik 1991.

Systemy ekspertowe w zarządzaniu

dokończenie ze str. 6

Prowadzone niejako kilkutorowo rozważania na temat zastosowań systemów ekspertowych w zarządzaniu, skłaniają do sformułowania pewnych ogólnych wniosków, a mianowicie:

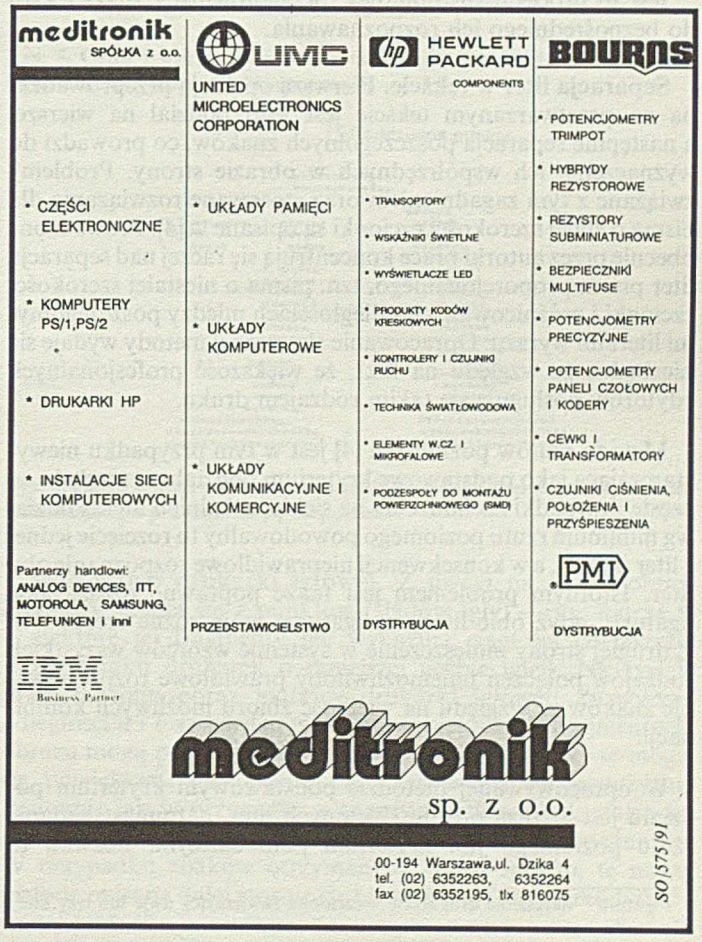
- sprowadzanie problematyki zarządzania do zbioru dobrze ustrukturalizowanych i programowych decyzji jest uproszczeniem mogącym wypaczać istotę kierowania przedsiębiorstwem;
- efektywne informatyczne wspomaganie zarządzania wymaga zarówno stosowania baz danych, jak i baz wiedzy, zaś baza modeli powinna obejmować modele: optymalizacyjne, prognostyczne i diagnostyczne, symulacyjne itp.;
- systemy ekspertowe nie są narzędziem uniwersalnym; do podstawowych dziedzin potencjalnego stosowania systemów ekspertowych w zarządzaniu należy zaliczyć wspomaganie:
  - negocjacji,
  - rozwiązywania sytuacji konfliktowych,
  - analiz marketingowych,
  - wyboru wariantu prowadzenia kampanii reklamowej,
  - wielokryterialnej analizy porównawczej wariantów rozwoju firmy,
  - diagnostyki organizacyjnej itp.;
- szczególną dziedziną zastosowań systemów ekspertowych jest sfera zarządzania strategicznego;
- w procesie tworzenia modeli systemów wspomagania zarządzania centralną rolę spełnia wybór modelu decydenta;
- cechy modelu decydenta w znacznym stopniu determinują wybór narzędzi komputerowego wspomagania, a w szczególności wybór systemu ekspertowego;
- nie istnieje uniwersalny model firmy, tak jak nie istnieje uniwersalny model decydenta, stąd wybór modelu firmy i modelu decydenta wymaga gruntownej analizy konkretnego przedsiębiorstwa i jego otoczenia;
- nie istnieją uniwersalne narzędzia tworzenia ekspertowych systemów wspomagania zarządzania (decyzji) i dlatego proces projektowania takich systemów jest procesem złożonym, wymagającym specjalistycznego komputerowego wspomagania projektowania;
- przyszłość w dziedzinie systemów wspomagania zarządzania należy do ekspertowych systemów wspomagania decyzji i negocjacji;
- aczkolwiek rozszerza się wachlarz środków komputerowego wspomagania zarządzania, to w centrum istotnych spraw dla każdego przedsiębiorstwa znajdują się ludzie ze swymi wartościami oraz indywidualnymi i grupowymi interesami, funkcjonujący w określonych strukturach organizacyjnych oraz dynamicznej info- i technosferze.

LITERATURA

[1] Bubnicki Z.: Wstęp do systemów ekspertowych. PWN, Warszawa 1990
[2] Carison E. D., Sprague R. H.: Building Effective Decision Support System. New Jersey 1982
[3] Findeisen W. (red.): Analiza systemowa
[4] Francuz P.: Założenia teorii rozwiązywania problemów Herberta A. Simona. Studium krytyczne. „Przegląd psychologiczny", nr 3, 1986
[5] Laccoca L.: Autobiografia. KiW, Warszawa, 1988
[6] Kłykow J. I., Jurek J. K.: Dialogowe semiotyczne systemy podejmowania decyzji. PWN, Warszawa 1988
[7] Kozielecki J.: Koncepcja transgresyjna człowieka. PWN, Warszawa 1987
[8] Łariczew O. I.: Nauka i iskusstwo priniatija reszenij. Nauka, Moskwa 1979
[9] Nosal Cz.: Psychologiczne modele umysłu. PWN, Warszawa 1990
[10] Nowak E.: Problem informacji w modelowaniu ekonometrycznym. PWN, Warszawa 1990
[11] Nowakowska M.: Teoria działania. PWN, Warszawa 1979
[12] Quade E. S.: Analysis for Public Decisions. RAND Corporation 1989
[13] Radzikowski W.: Komputerowe systemy wspomagania decyzji. PWE, Warszawa 1990
[14] Sienkiewicz P.: Inżynieria systemów. MON, Warszawa 1983
[15] Sienkiewicz P.: Inżynieria systemów kierowania. PWE, Warszawa 1988
[16] Sienkiewicz P.: Systemy kierowania. WP, Warszawa 1989.

Jedyną możliwością regularnego otrzymywania INFORMATYKI jest prenumerata.

Szczegóły na drugiej stronie okładki.



IWONA DĄBROWSKA
DOROTA DEPOWSKA
Instytut Elektrotechniki Teoretycznej
i Miernictwa Elektrycznego
Politechnika Warszawska

Podstawowe problemy rozpoznawania pisma drukowanego w systemach off-line

Podczas rozpoznawania znaków metodami off-line [2] pojawia się wiele problemów utrudniających właściwą ekstrakcję cech, a w rezultacie prawidłową identyfikację znaku. Trudności w rozpoznawaniu są związane z precyzją druku znaku, z szumami wprowadzanymi przez skaner oraz z poprawnością ułożenia kartki zawierającej tekst w skanerze. Poniżej zostaną omówione problemy rozpoznawania znaków dly systemu opisanego w [2], a składającego się z przetwarzania wstępnego oraz właściwego rozpoznawania, opartego na badaniu wklęsłości i obszarów zamkniętych.

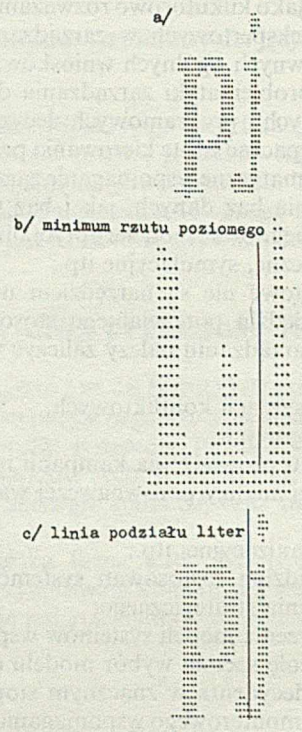
Wstępne przetwarzanie rozpoznawanego tekstu

Jest to proces dwustopniowy, przygotowujący znaki tekstu do bezpośredniego ich rozpoznawania.

Separacja liter w tekście. Pierwszą operacją przeprowadzaną na przetwarzanym tekście jest jego podział na wiersze, a następnie separacja poszczególnych znaków, co prowadzi do wyznaczenia ich współrzędnych w obrazie strony. Problemy związane z tym zagadnieniem oraz stosowane rozwiązania dla pisma o stałej szerokości czcionki są opisane w [4]. Prowadzone obecnie przez autorki prace koncentrują się raczej nad separacją liter pisma proporcjonalnego, tzn. pisma o niestałej szerokości czcionki i zróżnicowanych odległościach między poszczególnymi literami wyrazu. Opracowanie skutecznej metody wydaje się niezbędne ze względu na fakt, że większość profesjonalnych edytorów posługuje się takim rodzajem druku.

Metoda rzutów poziomych [4] jest w tym przypadku niewystarczająca jako podstawowe kryterium podziału ze względu na częste przypadki zachodzenia na siebie sąsiednich liter. Podział wg minimum rzutu poziomego powodowałby tu rozcięcie jednej z liter (rys. 1), a w konsekwencji nieprawidłowe rozpoznanie obu liter. Istotnym problemem jest także poprawne rozdzielanie ligatur[1)], gdyż obie litery w ligaturze mają różne szerokości. Z drugiej strony umieszczenie w systemie wzorców wszystkich rodzajów połączeń uniemożliwiłoby prawidłowe rozpoznawanie znaków ze względu na wielkość zbioru możliwych kombinacji.

W opracowywanej metodzie podstawowym kryterium podziału jest analiza profilu sklejonych liter, natomiast badanie rzutu poziomego jest kryterium pomocniczym. Pozwoli to uniknąć przypadkowego rozcinania liter. Ze względu na zmienną szerokość liter, każdy znak, którego szerokość jest większa od zadanej wartości (określonej statystycznie na podstawie badań) jest sprawdzany pod względem obecności cech, wskazujących na sklejenie. Cechy te muszą być dobrane bardzo dokładnie, gdyż wiele pojedynczych liter, np. *m, w* przechodzi przez ten próg klasyfikacji, ponieważ są szersze od niektórych ligatur, np. *fl, fi*. Prace te są na etapie testowania metody.

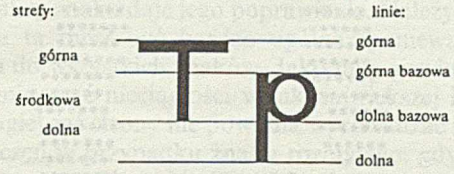


Rys. 1. Przykłady liter zachodzących na siebie:
a) obraz liter *pj* po skanowaniu
b) rzut poziomy tych liter
c) obraz pojedynczych liter po podziale wg minimum rzutu poziomego

Wstępna klasyfikacja znaków. Po wydzieleniu w tekście poszczególnych znaków następuje ich wstępna klasyfikacja na podstawie metody statystyczno-topologicznej [3]. Pozwala ona zaklasyfikować znak do jednej z siedmiu kategorii [2, 3] wg kryterium struktury słowa, zawierającego ten znak. W poszczególnych słowach wyodrębnia się trzy strefy: górną *U*, środkową *M* oraz dolną *L* (rys. 2). Współrzędne stref oblicza się na

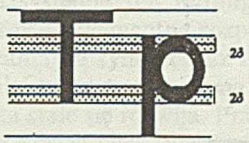
[1)] Ligatura – określenie drukarskie – czcionka zawierająca dwie lub trzy litery albo znaki.

podstawie współrzędnych dolnego i górnego rzędu piksli poszczególnych liter słowa. Położenie litery w stosunku do współrzędnych tych stref pozwala sklasyfikować znak do odpowiedniej kategorii.



Rys. 2. Podział słowa na sfery

Wysokość znaków należących do jednej kategorii nie jest stała, nawet dla tej samej litery, dlatego też określa się pewien przedział zmienności wysokości znaku δ, jako procentową część wysokości strefy M (rys. 3). Dodatkową trudność sprawiają tu typowo polskie litery alfabetu, np. *ą, l, ń,* które w wielu krojach czcionki mają niestandardową wysokość, bądź też są przesunięte do góry lub do dołu. Ich obecność z jednej strony bardzo utrudnia prawidłowe obliczenie szerokości stref, z drugiej zaś – prawidłową klasyfikację przesuniętych liter. W zależności od wielkości tych różnic istnieją dwa rozwiązania – bardzo precyzyjne dobranie współczynnika δ lub zaklasyfikowanie tych liter do odrębnej kategorii, spełniającej dodatkowe warunki. Opisana tu metoda wstępnej klasyfikacji, pomimo wymienionych trudności realizacyjnych, jest szczególnie przydatna dla alfabetu polskiego, który składa się z dużo większej liczby znaków (35) niż alfabet angielski (26), ponieważ trudniej jest w takim przypadku wybrać zbiór cech jednoznacznie określających jeden znak. Po dokonaniu wstępnej klasyfikacji, znaki są identyfikowane w mniejszych grupach, co znacznie ułatwia i przyspiesza proces rozpoznawania.



Rys. 3. Przedziały zmiennej wysokości liter

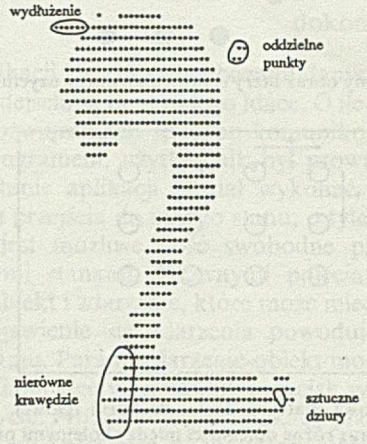
Ekstrakcja cech znaków

Jest ona utrudniona przez zakłócenia pochodzące z różnych źródeł.

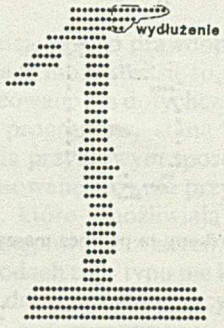
Szumy powstałe w wyniku skanowania rastrowego. Obraz literatury po skanowaniu rastrowym jest zaszumiony. Może być wydłużony, mieć oddzielne punkty lub sztuczne dziury, czy też nierówne krawędzie (rys. 4). Zakłócenia te utrudniają lub nawet uniemożliwiają określenie charakterystycznych cech dla danego znaku z powodu zmiany wyglądu jego obrazu wejściowego.

W opisywanym systemie podstawową cechą identyfikującą znak jest obszar zamknięty. W przypadku znaku cyfry 2 z rys. 4 okazałoby się, że ma on obszar zamknięty, choć nie jest to cecha identyfikująca ten znak. Znak z rys. 5 ma wyraźne wydłużenie w górnej części obrazu znaku, wskutek czego nie będzie on prawidłowo rozpoznany. Dopiero z kontekstu można byłoby stwierdzić, czy jest to cyfra 1 czy litera *l*. Na rys. 6 pokazano literę *n*. Jednakże pojawienie się zakłócenia nad jej właściwym obrazem (rozpatrzono najbardziej niekorzystną sytuację), powoduje, że litera ta będzie rozpoznana jako *ń*.

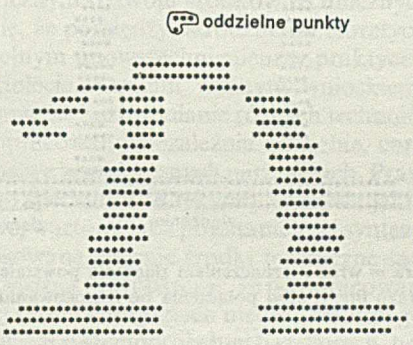
Z powyższych rozważań wynika, że niektórych zakłóceń nie można wyeliminować, ponieważ trzeba stosować ogólne reguły poprawiania obrazu znaku. Jednym z najczęściej pojawiających się zakłóceń są sztuczne dziury oraz niewielkie obszary zamknięte. Aby je wyeliminować, a przynajmniej częściowo zmniejszyć, autorki opracowały nową opisaną w dalszej części artykułu, metodę poprawy jakości obrazu znaku.



Rys. 4. Cyfra 2 wraz z oznaczeniem zakłóceń po procesie skanowania rastrowego

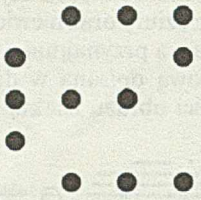


Rys. 5. Cyfra 1 wraz z oznaczeniem zakłócenia – wydłużenie – uniemożliwiającego prawidłowe jej rozpoznanie

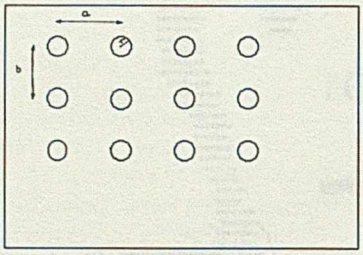


Rys. 6. Litera *n* wraz z oznaczeniem zakłócenia – oddzielne punkty – uniemożliwiającego prawidłowe jej rozpoznanie

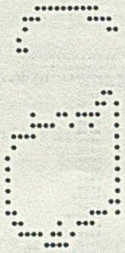
Stosowanie drukarki igłowej. Z uwagi na to, że obraz literatury składa się z punktów (śladów igły) – rys. 7 oraz 8 – niektóre jej fragmenty nie wykazują ciągłości (zwłaszcza w przypadku drukarek o mniejszej liczbie igieł). Na rys. 8 przedstawiono obraz matrycy drukarki igłowej. Zazwyczaj odległości *a* i *b* są bardzo małe, ale w niektórych fragmentach obrazu mogą pojawiać się wyraźne przerwy. Przerwy te mogą się powiększać, gdy taśma barwiąca uległa nieco zużyciu. Podobnie jak w przypadku sztucznych dziur, zniekształca to obraz litery i często uniemożliwia jej właściwe rozpoznanie. W przypadku znaków otrzymanych w ten sposób, tę nową metodę opisaną dalej stosuje się także, lecz w nieco zmodyfikowanej postaci.



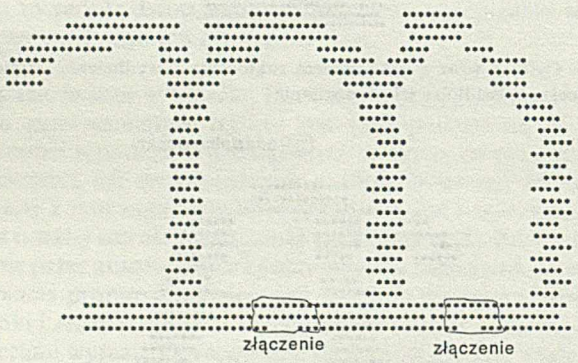
Rys. 7. Powiększony obraz litery *e*, otrzymany przy użyciu drukarki igłowej



Rys. 8. Przykładowa matryca druku drukarki igłowej, obrazująca brak ciągłości obrazu oraz różne odległości między kolejnymi punktami w pionie i w poziomie

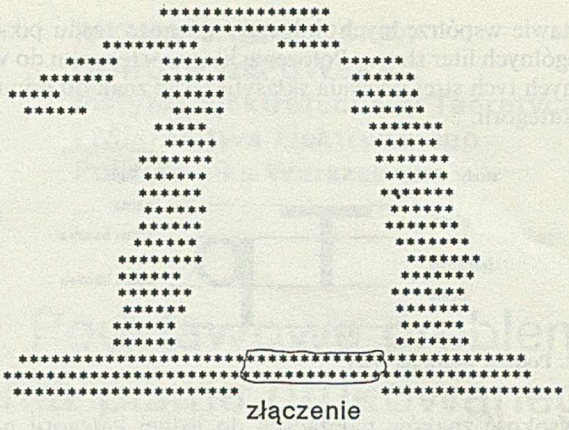


Rys. 9. Obraz litery *a*, uzyskany za pomocą maszyny z rozbitą czcionką



Rys. 10. Litera *m* wraz z oznaczeniem złączenia powstałego albo wskutek rozmycia obrazu litery, albo połączenia po zastosowaniu uzupełniania ze ścienianiem

Stosowanie maszyny z rozbitą czcionką. W tym przypadku litery są rozmazane i rozmyte, a intensywność tuszu nie jest jednakowa w całym obrazie znaku. Mogą tu pojawiać się dodatkowo dziury, których eliminacja przebiega analogicznie, jak w podstawowej wersji metody poprawiania obrazu. Rys. 9 przedstawia literę *a* uzyskaną z maszyny z rozbitą czcionką. Obraz tej litery wykazuje wyraźną przerwę, utrudniającą właściwe rozpoznanie. Z drugiej zaś strony, przy rozmyciu mogą pojawiać się nienaturalne połączenia w obrazie litery (dodatkowe obszary zamknięte) – rys. 10 oraz 11. Analiza litery *m* oraz *n* odpowiednio z rys. 10 i 11 wykazałaby, że znak z rys. 10 ma dwa poziome obszary zamknięte, a z rys. 11 – jeden, choć nie są to cechy charakteryzujące litery *m* oraz *n*. Prawidłowa identyfikacja takich znaków nie jest możliwa. Nie opłaca się też opracowywanie bloku uwzględniającego poszczególne możliwe zniekształcenia, gdyż cały system rozpoznawania liter rozrósłby

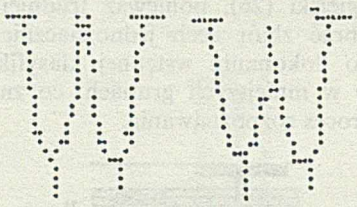


Rys. 11. Litera *n* wraz z oznaczeniem złączenia powstałego albo wskutek rozmycia obrazu litery, albo połączenia po zastosowaniu uzupełniania ze ścienianiem

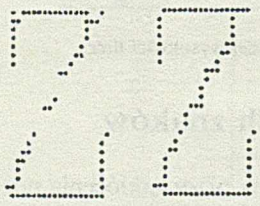
się znacznie, a i tak nie można przewidzieć wszystkich możliwych przypadków sklejeń. Dość rozsądnym sposobem eliminacji niewłaściwego rozpoznawania jest zastosowanie bloku uczącego.

Metody poprawy jakości obrazu znaku

Są one uzależnione od specyfiki źródeł powodujących zakłócenia.



Rys. 12. Litera *W* przed oraz po uzupełnieniu



Rys. 13. Litera *Z* przed oraz po uzupełnieniu

Podstawowa wersja metody eliminacji szumów skanowania rastrowego. Metoda poprawiania obrazu znaku wykorzystuje ścieśnianie oraz uzupełnianie brakujących piksli. Umożliwia ona eliminację sztucznych dziur i niewielkich obszarów zamkniętych. Znajduje zastosowanie w przypadku obrazów o dwóch stanach, tzn. zawierających piksle czarne (zapalone) oraz białe (niezapalone), tzn. nie uwzględnia poziomów szarości. Jako taka może być zastosowana w systemach OCR, w których litery składają się z czarnych piksli, a tło z białych. Wszystkie piksle w obrazie znaku (uzyskanym po ścienieniu pierwotnego obrazu) są otaczane pikslami, będącymi najbliższymi sąsiadami danego piksla (otoczenie ósemkowe [2]). Uzyskany w ten sposób nowy obraz znaku jest poddawany ponownemu ścienieniu. Efekty tej metody pokazano na rys. 12 oraz 13. Analizując dokładniej obrazy litery *W* (rys. 12), czy też litery *Z* (rys. 13) można zauważyć, iż oprócz usunięcia nieciągłości uzyskano lokalne poprawienie obrazu znaku (widoczne zwłaszcza w przypadku litery *W*). Pozornie wydaje się, że zmiana wyglądu litery po procesie poprawiania stanowi wadę opisywanej metody, ale

tak nie jest. Sam obraz znaku przed poprawianiem nie jest idealną kopią obrazu znaku wejściowego (tzn. z kartki papieru), a liczne testy wykazały iż powtórne ścienianie często wygładza kształt znaku. Z kolei czas potrzebny na uzupełnianie i ścienianie jest bardzo krótki w porównaniu z korzyściami, jakie dla dalszej analizy znaku daje jego poprawienie. Należy zaznaczyć, że metoda ta musi być bardzo ogólna, ponieważ jest ona stosowana do wszystkich znaków. Jako taka powinna z jednej strony eliminować nieciągłości w jak największej liczbie znaków, z drugiej zaś strony nie powinna wprowadzać niepotrzebnych połączeń w przypadku znaku rozmytego, gdy odległości pomiędzy poszczególnymi jego częściami są zmniejszone w nienaturalny sposób. Liczne testy wykazały, że liczba korzystnych efektów jest znacznie większa niż liczba niekorzystnych połączeń.

Metoda poprawy obrazu znaku drukarki igłowej. Idea tej metody jest taka sama, jak opisanej powyżej, z tym, że występuje tu niewielka modyfikacja. Znaki z drukarki igłowej są zaliczane do grupy znaków o złej jakości. W opracowywanym systemie istnieje dodatkowa opcja umożliwiająca obsługę tych znaków. Obraz znaku jest tylko raz ścieniany. Po wyodrębnieniu z tekstu jest on poddawany uzupełnianiu ósemkowemu, a dopiero potem – ścienianiu. Takie podejście umożliwia eliminację większych przerw, jakie pojawiają się w opisywanym przypadku.

* * *

Rozwiązanie zadania rozpoznawania znaków drukowanych jest problemem trudnym i złożonym ze względu na przypadkowe pochodzenie źródła błędów. Zachodzi tu konieczność stosowania metod wspomagających, które ułatwiają proces identyfikacji oraz znacznie poprawiają jego skuteczność, ale wydłużają czas jego wykonania. Opisana metoda poprawy obrazu znaków została wszechstronnie przetestowana oraz zastosowana w opracowywanym systemie IOCR (ang. *Intelligent Optical Character Recognition* – inteligentne rozpoznawanie znaków). Pomimo wprowadzania na rynek coraz to nowych systemów rozpoznawania znaków, są to programy wciąż niedoskonałe i dlatego dziedzina ta stale się rozwija. Prace badawcze prowadzone na Zachodzie przechodzą od systemów OCR (opierających się często na porównywaniu wzorców) do systemów IOCR, posługujących się technikami przypominającymi sposób rozpoznawania znaków dokonywanych przez człowieka. Proponuje się zaopatrywanie nowoczesnych systemów w bloki uczące i bloki rozpoznawania kontekstowego, korzystające ze słowników.

LITERATURA

[1] Bourbakis N. O.: A parallel-symetric thinning algorithm. Patters Recognition, Vol. 22, No. 4, 1989

[2] Dąbrowska I., Depowska D.: Metody automatycznego rozpoznawania znaków. Informatyka nr 4, 1992

[3] DeLuca P. G., Gisatti A.: Printed character preclassification based on word structure. Pattern Recognition, Vol. 24, No. 7, 1991

[4] Dołęga C. G.: Problemy wyszukiwania znaków w rozpoznawaniu tekstu drukowanego. Informatyka nr 5, 1991

[5] Kumar P., Bhatnager D., Rao P. S.: Pseudo one pass thinning algorithm. Pattern Recognition Letter Vol. 12, No. 9, 1991

[6] Porod W., Ferry D. K.: Pattern recognition in highly integrated circuits. Pattern Recognition, Vol. 18, No. 2, 1985

[7] Zawada A.: Automatyczne czytanie pisma. Informatyka nr 7, 1991.

Języki czwartej generacji a systemy CASE

dokończenie ze s. 12

tworzenia aplikacji związanych z bazami danych. Konsekwencje takiego podejścia są dość daleko idące. O ile w stosowanych do tej pory rozwiązaniach sposobu komunikowania się użytkownika z programem, użytkownik był prowadzony za rękę i w danym stanie aplikacji musiał wykonać określony ciąg działań w celu przejścia do innego stanu, o tyle przy podejściu obiektowym jest możliwe dość swobodne przenoszenie się między różnymi stanami. Głównymi pojęciami przy takim podejściu są obiekt i zdarzenie, które może mieć związek z tym obiektem. Pojawienie się zdarzenia powoduje odpowiednią reakcję programu. Parami zdarzenie-obiekt mogą być na przykład pole i zmiana jego wartości lub przycisk systemu komunikacji i jego użycie. Programowanie aplikacji przy takim podejściu nie polega, jak do tej pory, na tworzeniu hierarchii funkcji do wykonania w sposób sekwencyjny, a raczej na przewidywaniu i opisywaniu wszystkich możliwych zdarzeń, jakie mogą wystąpić w systemie.

Konsekwencją takiej, bardzo prawdopodobnej, zmiany filozofii tworzenia aplikacji może stać się to, że dobrze opanowane metody CASE, opracowane dla dotychczasowego trybu projektowania i rozwoju programów, staną się bezużyteczne lub niewygodne do użycia przy nowym sposobie pracy. Istnieją co prawda techniki, stosowane głównie przy tworzeniu systemów czasu rzeczywistego, które umożliwiają projektowanie systemów opartych na występowaniu zdarzeń. Niestety w opracowanych do tej pory metodach tego typu nie kładzie się zbyt dużego nacisku na problem danych występujących w systemie.

Dotychczasowa historia rozwoju narzędzi informatycznych prowadzi do kilku interesujących wniosków. Wbrew przekonaniu o błyskawicznym rozwoju środków technicznych informatyki okazuje się, że pomiędzy narodzinami teoretycznych koncepcji, a ich pełnym upowszechnieniem w praktyce mijają niekiedy dziesięciolecia. Drugim ważnym wnioskiem jest wzajemne przenikanie się i uzupełnianie różnych technologii. CASE, 4GL i GUI, opracowane niezależnie od siebie, coraz bardziej łączą się ze sobą we współczesnych narzędziach. Prawdopodobnie już niedługo specjaliści od reklamy wymyślą jakiś termin na określenie nowej wartości jaka powstanie z tej syntezy. Wreszcie po trzecie, stosowane obecnie środki techniczne są dalekie od doskonałości. Można przypuszczać, że po opracowaniu i zintegrowaniu w przyszłych narzędziach metod graficznego projektowania obiektowych systemów obsługi baz danych, będzie trzeba coś zrobić z tak zwanym kryzysem komunikacji. Głównym problemem współczesnych twórców systemów informatycznych staje się, wobec coraz bardziej rosnącej złożoności tych systemów, zrozumienie rzeczywistych potrzeb już nie pojedynczego użytkownika, a całych organizacji. Można się spodziewać, że kolejnymi składnikami, które będą włączane do powszechnego instrumentarium informatycznego, będą elementy technologii systemów doradczych (ang. *expert system*).

TIFF standard grafiki pikselowej

dokończenie ze s. 17

LITERATURA

[1] Bielecki J.: Grafika Turbo...

[2] Kernighan B. W.: Język C

[3] Kozdrowicz T.: IBM PC i PC DOS

[4] Lindley C. A.: Practical Image Processing in C

Algorytmika – sedno informatyki

Kto czytał książkę Cerama „Bogowie, groby i uczeni" pamięta zapewne, jak bardzo przykuwała ona uwagę czytelnika. Ta fascynująca opowieść o największych odkryciach archeologicznych sprawiła, że wiele osób zaczęło interesować się dziejami starych kultur, a są też i tacy, którzy dzięki niej odkryli w sobie powołanie archeologa. Przyczyną tego zainteresowania nie jest bynajmniej atrakcyjność samej archeologii, której codzienność to żmudna, mrówcza praca, ale sposób, w jaki Ceram przedstawił jej najbardziej spektakularne osiągnięcia.

Książka ta ma jeszcze inną zaletę. Można powiedzieć, że ona nie starzeje się. Tak samo jak była atrakcyjna w chwili pierwszego wydania, jak jest atrakcyjna dziś, tak będzie zapewne przyciągała czytelników jeszcze przez wiele lat, chyba że nastąpią nowe odkrycia przyćmiewające swą sławą odkrycie Myken, Troi czy Babilonu.

Zapewne można wskazać równie fascynujące książki z dziedziny chemii, fizyki czy biologii. Wiadomo, że osiągnięcia tych przez wieki rozwijających się nauk są ogromne i wspaniałe, nietrudno więc pisać o nich wspaniałe książki.

Czy można jednak równie fascynująco pisać o informatyce i to w ten sposób, żeby przykuć uwagę czytelnika nie tylko dziś, ale również za dziesięć, czy dwadzieścia lat. Niektórzy są skłonni twierdzić, że nie ma czegoś takiego, jak czysta informatyka – teorią zajmują się matematycy, sprzęt robią elektronicy, a programowaniem parają się rzemieślnicy. Nawet jeśli uznamy, że informatyka może już dziś być uważana za odrębną dziedzinę nauki, to powstaje pytanie, czy w tej tak szybko zmieniającej się dziedzinie jest coś co może stanowić jej nieprzemijające sedno. Spójrzmy na naszą podręczną biblioteczkę informatyczną. Czy są w niej książki wydane dziesięć czy piętnaście lat temu? A jeśli tak, to czy zawierają one potrzebne nam dziś informacje? Zapewne za dziesięć lat wyrzucimy do kosza używane dziś podręczniki MS-DOS, MS-WINDOWS, Lotusa czy Clippera.

I oto pojawia się na naszym rynku wydawniczym książka Davida Harela „Rzecz o istocie informatyki Algorytmika"*). Nie jest to zwyczajna książka, jeszcze jeden podręcznik jakiejś tam algorytmiki. Jest to książka niezwykła – fascynująca opowieść, którą można czytać do poduszki, jedna z tych książek, po przeczytaniu której pozostaje żal, że się już skończyła.

Autor David Harel jest profesorem informatyki w Instytucie Weizmanna, w Izraelu. Zyskał szerokie uznanie za swe prace badawcze w różnych dziedzinach informatyki, głównie w teorii złożoności i zagadnieniach współbieżności. Na przełomie lat 1984–1985 David Harel miał cykl pogadanek w „Galei Zahal", jednym z narodowych programów radiowych w Izraelu. Trudno sobie wyobrazić, jak można mówić interesująco o informatyce przez radio, nie pokazując tych wspaniałych możliwości komputerów, którymi chwalą się producenci sprzętu i oprogramowania na każdych targach komputerowych, takich jak CeBIT czy Infosystem. Harel jednak nie mówił o najnowszych osiągnięciach techniki komputerowej. Mówił o tym, co dla informatyki podstawowe i niezmienne – mówił o algorytmice.

Algorytmika, to studiowanie algorytmów. Natomiast algorytm to abstrakcyjny przepis opisujący proces, który może być wykonany przez człowieka, komputer lub dowolne inne urządzenie. Jest więc to pojęcie bardzo ogólne, choć najbardziej interesują nas oczywiście te algorytmy, które może realizować komputer.

Książka, o której tu mowa, powstała właśnie na podstawie radiowych pogadanek. Jest napisana językiem prostym, zrozumiałym dla laika, nie ma w niej żadnych skomplikowanych wzorów. Nie uczy programowania ani obsługi komputerów, a mimo to na jej podstawie można napisać wiele niebanalnych programów. Jest to przede wszystkim książka o algorytmice, ale ponieważ algorytmika jest rdzeniem, pniem całej informatyki, siłą rzeczy mówi ona o tym, co z tego pnia wyrasta. Wystarczy zajrzeć do skorowidza. Znaleźć tam można takie hasła, jak: języki programowania, współbieżność, bazy danych, bazy wiedzy, sztuczna inteligencja, kryptografia, systemy czasu rzeczywistego, systemy rozproszone.

Książka jest podzielona na cztery części, z których każda składa się z trzech rozdziałów. Razem mamy więc dwanaście rozdziałów plus przedmowa, posłowie i uwagi bibliograficzne. Część pierwsza mówi o rzeczach podstawowych: co to jest i jak działa komputer, jakie są jego ograniczenia, co to jest algorytm, jakie są jego cechy, omawia różne typy struktur danych i instrukcji sterujących oraz podaje przegląd najbardziej popularnych języków programowania. Część druga jest poświęcona metodologii tworzenia algorytmów, sposobom ich usprawniania oraz konieczności dowodzenia ich poprawności. Na tym etapie kończy się zazwyczaj wiedza i zainteresowania zwykłego programisty. Zwykle postępuje on bowiem według następującego schematu: jest problem, trzeba dla niego wymyśleć algorytm, następnie napisać program, a jeśli okaże się, że program „chodzi" za wolno, trzeba go umiejętnie zoptymalizować. Odpowiednio intensywne testowanie zapewnia nas o poprawności programu. Książka Harela zwraca uwagę na mankamenty takiego postępowania i pokazuje, że tworzenie naprawdę dobrego oprogramowania nie jest jedynie rzemiosłem, lecz wymaga dużej wiedzy oraz zaangażowania zaawansowanego aparatu matematycznego.

W prawdziwie fascynujący świat algorytmiki wprowadzają nas dwie pozostałe części książki. W części trzeciej dowiadujemy się, że są problemy, do których rozwiązania nasze komputery są zdecydowanie za wolne, co więcej, nawet ich stu-, czy tysiąckrotne przyspieszenie nic nie da. Nie są to wbrew pozorom jakieś abstrakcyjne, wydumane przez teoretyków problemy, lecz bardzo konkretne zadania. Przykładowo, takim problemem jest znalezienie odpowiedzi na pytanie, czy z 25 klocków o różnokolorowych ścianach można ułożyć kwadrat tak, aby stykały się one ścianami tego samego koloru. Rozwiązanie problemów tego typu może zająć miliony, czy miliardy lat,

*) David Harel: Rzecz o istocie informatyki. Algorytmika. Wydawnictwa Naukowo-Techniczne, Warszawa 1992. Jest to tłumaczenie książki „Algorithmies. The Spirit of Computing" tego autora wydanej w 1987 r. przez wydawnictwo Adison-Wesley.

a więc są one dla nas w praktyce nierozwiązywalne. Nie jest to optymistyczna wiadomość, ale tak jak gotowi jesteśmy pogodzić się z myślą, że nie można przekroczyć prędkości światła, możemy również pogodzić się z tym, że rozwiązania niektórych problemów, leżą dla nas „za daleko". Gdy już się z tą myślą pogodzimy, autor ukazuje nam cały świat problemów, dla których nie potrafimy, a co gorsza nigdy nie będziemy potrafili, podać żadnego algorytmu. Fakt, że istnieją takie problemy ma wymowę filozoficzną, dotyczy bowiem granic poznania człowieka. I znów nie są to jakieś wydumane problemy, lecz np. bardzo praktyczny problem znalezienia algorytmu, który dla każdego programu potrafiłby stwierdzić, czy zatrzyma się on dla podanych danych. Harel przekonuje nas, że ta niemożność jest uniwersalna i nie zależy od używanego języka programowania, czy zastosowanego komputera.

Ostatnia część książki ukazuje nam nowe możliwości, jakie otwierają się przed algorytmiką, gdy dopuścimy współbieżność i zrezygnujemy z całkowicie poprawnych algorytmów na rzecz algorytmów poprawnych z prawdopodobieństwem bliskim jedności. Współbieżność daje ogromne możliwości – autor podaje nam przykład z konkursu budowy domków jednorodzinnych na czas, w którym zwycięski zespół zrealizował zadanie w niewiele ponad cztery godziny. I znów po pokazaniu nam, jak to współbieżność znakomicie przyspiesza realizację takich zadań, jak sortowanie, wyszukiwanie, czy mnożenie macierzy przez wektor, autor zaskakuje nas informacją o tym, że mimo zastosowania współbieżności to, co było bez niej praktycznie nierozwiązywalne nadal pozostaje takim również z nią.

O algorytmach probabilistycznych dowiadujemy się, że znakomicie przyspieszają realizację niektórych zadań. Na przykład, sprawdzenie tradycyjną metodą, czy jakaś bardzo duża (powiedzmy ponad stucyfrowa) liczba naturalna jest liczbą pierwszą, okazuje się być bardzo długie. Dzięki zastosowaniu algorytmu probabilistycznego wystarczy kilkadziesiąt lub kilkaset losowo dobranych sprawdzeń. Ale i probabilizm nie umożliwia nam rozwiązania problemów, dla których nieprobabilistyczne algorytmy realizują się nierozsądnie długo.

Ciekawych rzeczy można dowiedzieć się z tej książki o kryptografii, czyli o metodach szyfrowania danych. Okazuje się, że najlepsze metody są praktycznie nie do złamania dlatego, że istnieją problemy, których nie można rozwiązać w rozsądnym czasie. Jest to piękny przykład, jak można z pożytkiem wykorzystać niekorzystny skądinąd wynik badań uczonych.

W ostatnim rozdziale książki rozważa się możliwości osiągnięcia tego, do czego tak naprawdę informatyka cały czas dąży – stworzenia komputera na wzór i podobieństwo człowieka. Nie chodzi tu oczywiście o wygląd zewnętrzny, lecz o sztuczną inteligencję. Autor przekonuje nas, że komputery nigdy nie będą potrafiły myśleć, podobnie jak łodzie podwodne nigdy nie będą potrafiły nurkować, bo tak jak nurkowanie jest umiejętnością przypisywaną ludziom i zwierzętom, tak myślenie jest wyłącznym atrybutem człowieka. Na dowód podaje różne przykłady pytań, jakie można by zadawać takiemu inteligentnemu komputerowi i analizuje możliwość (a właściwie niemożliwość) rozsądnego na nie odpowiedzenia.

Rozdział kończy krótka, acz zabawna analiza możliwości komputerowego rozpoznawania mowy. Czy komputer ma szanse odróżnić zdanie *Umyj nogi i ręce* od zdania *Umyj nogi Irence*? Czy jeśli powie się komputerowi, że na stole leży chleb w koszyku i sałatka na talerzu, a potem mu się powie, że Jan z trudem zjadł w końcu to wszystko, to czy będzie on wiedział, co zjadł Jan? Chleb, sałatkę, może także koszyk i talerz, a może nawet stół?

Abyśmy już nie mieli żadnych wątpliwości, autor przytacza cztery krótkie wymyślne dialogi przeprowadzane przez telefon między człowiekiem a wyimaginowanym inteligentnym komputerowym sekretarzem. Choćby dla tych językowych gagów warto kupić i przeczytać tę książkę.

Wszystkie omawiane w książce zagadnienia są zilustrowane krótkimi, choć wcale niebanalnymi, przykładami. Nie posługując się żadnym konkretnym językiem programowania, a używając jedynie języka naturalnego, Harel podaje i szczegółowo wyjaśnia wiele interesujących algorytmów mających duże znaczenie w różnych gałęziach zastosowań informatyki. Poznajemy więc różne algorytmy wyszukiwania, sortowania, znajdowania drogi w grafie, szyfrowania, probabilistycznego badania, czy dana liczba jest liczbą pierwszą, probabilistycznego wyszukiwania wzorca, synchronizacji dostępu do sekcji krytycznej, współbieżnego mnożenia macierzy przez wektor, rozgrywania partii pokera przez telefon, rozwiązania problemu wież z Hanoi, problemu pięciu filozofów. Zapoznajemy się także z wieloma poważnymi problemami algorytmicznymi, których rozwiązań autor nie podaje, gdyż albo są zbyt skomplikowane, albo po prostu nie istnieją. Są to na przykład: problem kolorowania map i grafów, problem stwierdzania prawdziwości formuł logicznych, problem komiwojażera, czyli znajdowanie optymalnej ścieżki w grafie, problem różnych układanek, problem stopu, problem sprawdzenia, czy dwie gramatyki opisują ten sam język itp.

Każda część i rozdział są opatrzone cytatami z Biblii. Cytaty zamieszczone w oryginale są wzięte z tłumaczenia Biblii na angielski, jakiego dokonał ojciec autora Harold Fisch. Trzeba przyznać, że cytaty te są dobrane bardzo tendencyjnie. Czasami jest to tylko kilka słów wyrwanych z kontekstu, które w kontekście książki nabierają nowego, nieraz zupełnie innego znaczenia. W polskim tłumaczeniu użyto cytatów z Biblii księdza Wujka. Niestety w niektórych przypadkach zniknęła dwuznaczność słów angielskich i pozostało nie to znaczenie, które chciał uwypuklić Harel.

Kontrowersyjny jest sposób przedstawienia odwołań do literatury. Autor przytacza w swej książce wyniki pracy wielu uczonych. Z nazwiska wymienia tylko kilku nieżyjących już prekursorów informatyki, takich jak Turing, Church, Hollerith. W książce popularno-naukowej jest to oczywiście dopuszczalne. Zbyt częste powoływanie się na autorów wyników mogłoby przytłumić myśl przewodnią książki. Dlatego dobrym pomysłem jest zamieszczenie uwag bibliograficznych w oddzielnym rozdziale na końcu książki. Jednak odsyłając nas do fachowych pism i książek Harel tylko w niektórych przypadkach wyraźnie wskazuje autorów omawianych przez niego teorii czy algorytmów. W wielu przypadkach jedynie po nazwiskach autorów cytowanych pozycji można domyślać się, kto co tak naprawdę zrobił.

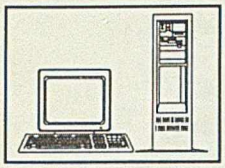
Jak widać, kontrowersje dotyczą raczej spraw pozamerytorycznych. Zresztą wielkie dzieła na ogół bywają kontrowersyjne. Nowe, świeże ujęcie musi zrywać ze starymi kanonami. Książka Harela jest właśnie takim świeżym spojrzeniem na informatykę, spojrzeniem ukazującym istotę tej młodej jeszcze dziedziny nauki. Dlatego powinni ją przeczytać wszyscy. Naukowcom daje ona lepszą orientację w tych dziedzinach, którymi nie zajmują

się bezpośrednio, oraz pokazuje, jak jasno i prosto można mówić o swojej dziedzinie. Tak zwanym profesjonalistom uświadamia granice ich możliwości; mogą oni w niej znaleźć wiele ciekawych problemów i niebanalnych algorytmów. Laikom pozwala zrozumieć, czym naprawdę jest informatyka, skąd bierze się pozorna inteligencja komputera. Dla wszystkich jest to książka o rozwoju ludzkiej myśli, o tym, jak człowiek powoli odkrywa i uświadamia sobie niemożność rozwiązania pewnych problemów niezależnie od tego, jaką wspaniałą techniką teraz dysponuje i jaką będzie dysponował w przyszłości.

ZBIGNIEW WEISS
Instytut Informatyki UW

Sienkiewicz P.: Systemy ekspertowe w zarządzaniu INFORMATYKA 1993, nr 2, s. 1 Aktualny stan wiedzy o procesach decyzyjnych oraz realne możliwości zastosowania systemów ekspertowych w zarządzaniu.	**Sienkiewicz P.: Expert systems in management** INFORMATYKA 1993, No. 2, p. 1 Actual knowledge on decision processes and realistic possibilities for application of expert systems in management.	**Sienkiewicz P.: Expertensysteme in Verwaltungsbereich** INFORMATYKA 1993, Nr. 2, S. 1 Eine Charakteristik von heutigem Wissen über Entscheidungsprozesse und realistische Möglichkeiten von Expertensystemeanwendung in Verwaltungsbereich.
Kisielnicki J.: Informacyjna infrastruktura zarządzania – potrzeby i tendencje zmian INFORMATYKA 1993, nr 2, s. 8 Ewolucja informacyjnej infrastruktury zarządzania w nawiązaniu do aktualnej sytuacji krajowej.	**Kisielnicki J.: The informative management infrastructure – needs and tendency of changes** INFORMATYKA 1993, No. 2, p. 8 Evolution of the informative management infrastructure with reference to actual home situation.	**Kisielnicki J.: Informationsinfrastruktur der Verwaltung – Bedürfnisse und Änderungstendenzen** INFORMATYKA 1993, Nr. 2, S. 8 Eine Evolution von Informationsinfrastruktur der Verwaltung mit Anknüpfung zur inländischen Situation.
Fuglewicz P.: Języki czwartej generacji a systemy CASE INFORMATYKA 1993, nr 2, s. 11 Popularne omówienie podobieństw oraz ewolucji języków czwartej generacji i systemów CASE.	**Fuglewicz P.: Fourth generation languages and CASE systems** INFORMATYKA 1993, No. 2, p. 11 Popular discussion of 4GL and CASE systems similarity and development.	**Fuglewicz P.: Sprachen der vierten Generation und CASE-Systeme** INFORMATYKA 1993, Nr. 2, S. 11 Eine populäre Besprechung von Ähnlichkeiten und Evolution der Sprachen von der 4. Generation und der CASE-Systeme.
Bendig-Wielowiejski A., Ruciński A., Sarad J.: Zastosowania wielofunkcyjnych elektronicznych kart bezstykowych INFORMATYKA 1993, nr 2, s. 13 Ogólna charakterystyka oraz zalety stosowania wielofunkcyjnych elektronicznych kart bezstykowych.	**Bendig-Wielowiejski A., Ruciński A., Sarad J.: Multifunctional electronic contactless card applications** INFORMATYKA 1993, No. 2, p. 13 Characteristics of multifunctional electronic contactles card and advantages of its application.	**Bendig-Wielowiejski A., Ruciński A., Sarad J.: Anwendungen von elektronischen kontaktlosen Mehrfunktionskarten** INFORMATYKA 1993, Nr. 2, S. 13 Eine Charakteristik und Vorteile der Anwendung von elektronischen kontaktlosen Mehrfunktionskarten.
Berent J. A.: TIFF – standard grafiki pikselowej INFORMATYKA 1993, nr 2, s. 16 Zwięzła charakterystyka standardu grafiki pikselowej TIFF.	**Berent J. A.: TIFF – a standard for pixel graphics** INFORMATYKA 1993, No. 2, p. 16 Concise characteristics of the TIFF standard for pixel graphics.	**Berent J. A.: TIFF – ein Standard für Pixelgraphik** INFORMATYKA 1993, Nr. 2, S. 16 Eine kurzgefasste Charakteristik des TIFF-Standardes für Pixelgraphik.
Bochiński J.: Matematyka w tworzeniu oprogramowania? INFORMATYKA 1993, nr 2, s. 18 Uzasadnienie wiodącej roli matematyki w tworzeniu oprogramowania.	**Bochiński J.: Mathematics in software creation?** INFORMATYKA 1993, No. 2, p. 18 Reason of mathematics leading part in software creation process.	**Bochiński J.: Mathematik in Herstellung von Software?** INFORMATYKA 1993, Nr. 2, S. 18 Eine Begründung der führenden Rolle von Mathematik bei der Softwareherstellung.
Dąbrowska I., Depowska D.: Podstawowe problemy rozpoznawania pisma drukowanego w systemach off-line INFORMATYKA 1993, nr 2, s. 22 Omówienie najważniejszych problemów występujących w procesie rozpoznawania pisma drukowanego w systemach off-line.	**Dąbrowska I., Depowska D.: Basic problems of printed character recognition in off-line systems** INFORMATYKA 1993, No. 2, p. 22 Discussion of main problems which appear by printed character recognition process in off-line systems.	**Dąbrowska I., Depowska D.: Grundprobleme der Druckschrifterkennung in Off-Line-Systemen** INFORMATYKA 1993, Nr. 2, S. 22 Eine Besprechung von wichtigsten Problemen, die bei Druckschrifterkennung in den Off-Line-Systemen auftauchen.

JUNISOFTEX®



Przedsiębiorstwo Produkcyjno-Usługowe Innowacji Technicznych Biuro Techniczno-Handlowe 44-100 Gliwice ul. Konstytucj 11.
tel. 31-57-00, 31-51-31, 31-75-10, fax 31-75-10, centr. 31-90-81 do 88, telex 036233
Zespół wdrożeń i analiz systemów komputerowych wewn. 15 ,25, 282, Zespół inżynierii oprogramowania wewn. 14, 20, 245, Zespół d/s serwisu systemów Novell i sprzętu komputerowego wewn. 16, 245, Dział Handlowy wewn. 18, 28, 201, Szkoła Komputerowa wewn. 282, Sekretariat wewn. 250, Prezes wewn. 13, 272, Dyrektor wewn. 12, 272, Główny Księgowy wewn. 256,
** Konto: PKO I/O Gliwice 27661-225298-136 **

OFERUJEMY

Kompleksową komputeryzację przedsiębiorstw obejmująca dostawę i wdrożenie zintegrowanych systemów komputerowych takich jak system:

finansowo-kosztowy
(automatyczne rozliczanie kosztów)
+
gospodarka materiałowa
+
środki trwałe
+
wyposażenie
+
obrót towarowy
fakturowanie sprzedaży
+
kadry-płace
+
techniczne przygotowanie produkcji

Sprzęt komputerowy w tym:

Komputery
Monitory
Dyski twarde
Drukarki
(mozaikowe i laserowe)
osprzęt sieciowy
(karty sieciowe, kable zwykłe i światłowodowe)

Systemy i oprogramowanie licencjonowane

DOS
Novell
Btrieve
SQL Base
Windows

Szkolenia komputerowe

podstawy obsługi komputerów
arkusze kalkulacyjne (Qpro, Lotus, EXCEL)
edytory tekstów (WordPerfect, TAG, ChiWriter)
bazy danych (dbase IV, Clipper, Paradox)
systemy operacyjne (DOS, Novell)

SOFTWARE SUPPORT FOR DOS, NOVELL 2.2, 3.11, 4.0, SQL BASE firmy GUPTA

*

OPROGRAMOWANIE DLA PRZEDSIĘBIORSTW

oferujemy własnej produkcji zakładowy system informatyczny PHU-Perfect(N) obejmujący swoim zakresem:
system finansowo-kosztowy umożliwiający automatyczne rozliczanie kosztów, atomatyczną dekretację, analizę rozrachunków
system obrotu materiałami, wyrobami, towarami (kody paskowe) w tym: stany magazynowe, ewidencje zamówień, fakturowanie sprzedaży (VAT) z automatyczną dekretacją rozdzielników kosztów i sprzedaży do systemu finansowo-kosztowego,
system ewidencji majątku trwałego i wyposażenia
system kadrowo-płacowy (SQL) z automatyczną dekretacją rozdzielników płacowych do systemu finansowo-kosztowego
system technicznego przygotowania produkcji (uk. 1993 r)
(oprogramowanie w systemie SQL BASE w przygotowaniu)

*

OPROGRAMOWANIE DLA URZĘDÓW MIAST I GMIN

oferujemy własnej produkcji system finansowo-podatkowy UMG-Perfect(N) obejmujący swoim zakresem:
system finansowo-księgowy,
systemy podatkowe obejmujące: ustalanie wymiarów, naliczanie i kontrolę płatności należności podatkowych dla dowolnych podatników i/lub płatników podatków na ich kontach i kontach dochodów ze swobodną korektą dowolnych informacji w dowolnym okresie a w tym:
ewidencja zapłat z wykorzystaniem kodów paskowych, emisja wezwań do zapłaty, naliczanie odsetek.
system ewidencji majątku

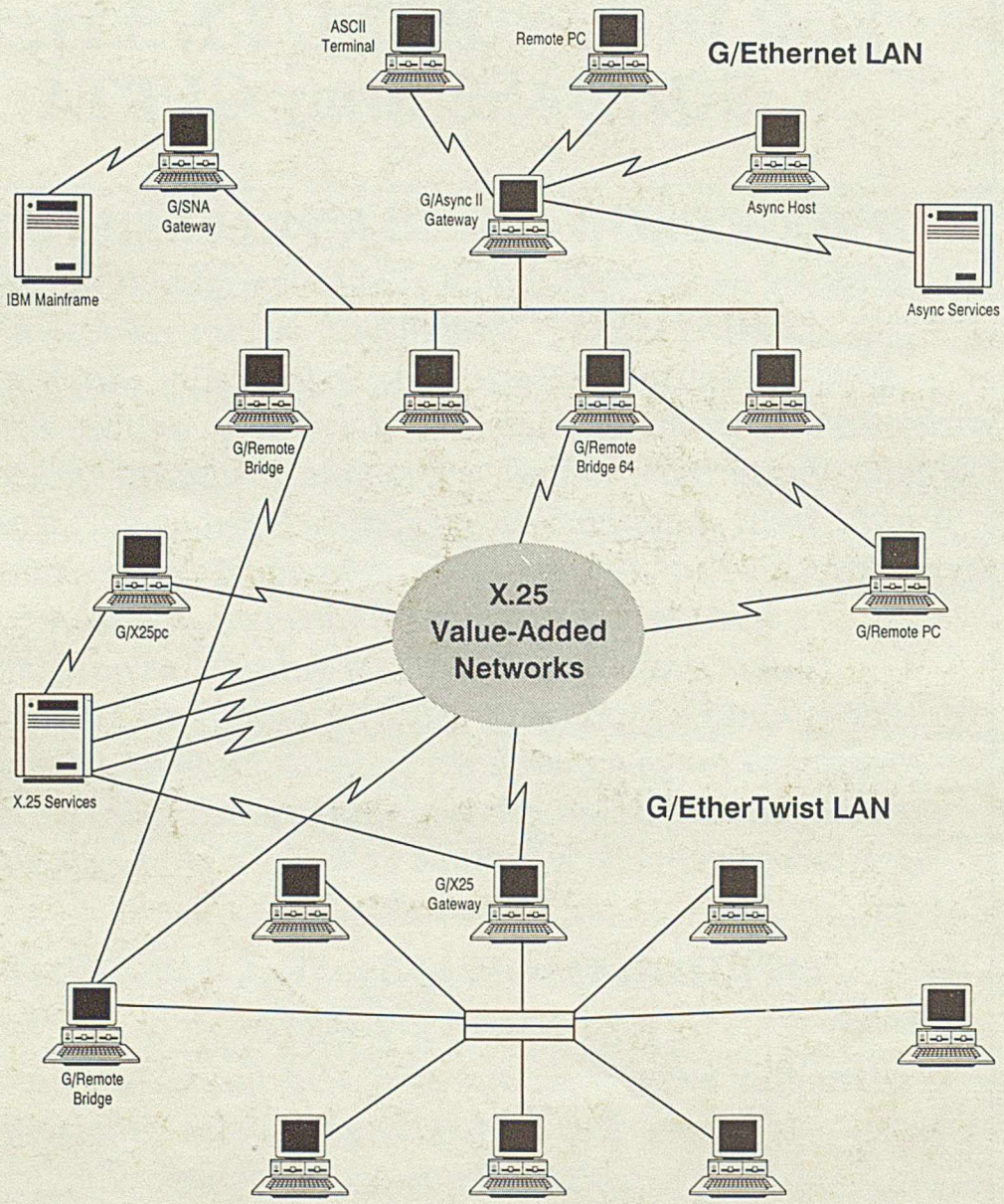
*

Zapraszamy do współpracy w zakresie wdrożen naszych systemów partnerów, - firmy informatyczne z całego kraju !!!

0/7/92

Produkty sieciowe LAN i WAN

Gateway communications, inc.



BEZKONKURENCYJNE PRODUKTY
SIECI KOMPUTEROWYCH NAGRADZANE PRZEZ:
PC MAGAZINE, LAN MAGAZINE, INFO WORLD
OFERUJE AUTORYZOWANY DYSTRYBUTOR:

MIKROB

PC MAGAZINE EDITORS' CHOICE January 31, 1989 G/Ethernet AT

PC MAGAZINE EDITORS' CHOICE October 16, 1990 G/EtherTwist

P.W.P.T. MIKROB SP. Z O.O.

20-346 Lublin, ul. Długa 5
Tel. (0-81) 420-61, faks 415-43, teleks 643776 MIKRO PL

0/15/91

Na życzenie wysyłamy katalog produktów. Atrakcyjny program współpracy dla dealer'ów.